

ORGANIZING FOR
EFFECTIVE SOFTWARE DEVELOPMENT

INPUT

ABOUT INPUT

THE COMPANY

INPUT provides planning information, analysis, and recommendations to managers and executives in the information processing industries. Through technology forecasting, analysis, INPUT supports in making informed decisions. Services are provided to users of computers, communications products and services.

The company carries out depth research. Working with clients on important issues, members analyze and interpret data, then develop recommendations. Innovative ideas to meet client needs. Clients receive reports. Access to data on which they can rely and continuous consulting.

Many of INPUT's professionals have nearly 20 years experience in areas of specialization. senior management positions in marketing, or planning enables INPUT to supply solutions to complex business problems.

Formed in 1974, INPUT is a leading international consulting firm. Clients include over 100 of the largest and most technically advanced companies.

UNITED STATES, West Coast

2471 East Bayshore Road
Suite 600
Palo Alto, California 94303
(415) 493-1600

Y-GTE
SUS

Way

Mantell, Stan

AUTHOR

Organizing for Effective

TITLE Software Development.

10/80

DATE

BORROWER'S NAME

California 90291

Coast

sey 07662

or)

erriwa Street
I.S.W. 2072

Italy
Milan 284-2850

JAPAN

INPUT Japan
Suite 1106
7-7-26 Nishi-Shinjuku
Tokyo
Japan 160
(03) 371-3082

ORGANIZING FOR
EFFECTIVE SOFTWARE DEVELOPMENT

X

Prepared For:
GTE SERVICES CORPORATION

INPUT LIBRARY

OCTOBER 1980

ORGANIZING FOR EFFECTIVE SOFTWARE DEVELOPMENT

TABLE OF CONTENTS

	<u>Page</u>
I INTRODUCTION	1
II EXECUTIVE SUMMARY	5
A. Major Findings And Conclusions	5
1. Effective Software Organizations	5
2. Software Development And Corporate Organization	6
a. Reporting Structure	6
b. Interdepartmental Relationships	8
c. Organizational Changes	8
3. Cost And Financial Ratios	9
4. Programming Standards	9
5. Quality Control	9
6. Measures Of Software Development Success	10
7. Future Of Software Development	11
B. Recommendations	12
1. Organizing For Effective Software Development	12
a. Reporting Structure	12
b. Interdepartmental Relationships	13
c. Organizational Changes	14
2. Cost And Financial Ratios	15
3. Programming Standards	15
4. Quality Control	15
5. Measure Of Software Development Success	15
6. Future Of Software Development	16
III ANALYSIS OF SOFTWARE DEVELOPMENT EFFORT	17
A. Evaluation Technique	17
B. Methodology Of Testing Ratings	20
C. Categorization Of Companies Interviewed	30
IV ORGANIZING FOR SOFTWARE DEVELOPMENT	35
A. Typical Reporting Structures Of Companies Surveyed	35
B. Interdepartmental Relationships	53
C. Organizational Changes	56
V SOFTWARE DEVELOPMENT COST AND FINANCIAL RATIOS	61

	<u>Page</u>
VI STANDARDS AND QUALITY CONTROL IN SOFTWARE DEVELOPMENT	71
A. Programming Standards	71
B. Quality Control	74
C. Measures Of Software Development Success	83
VII THE FUTURE OF SOFTWARE DEVELOPMENT	89
APPENDIX A: INTERVIEW PROFILES	93
APPENDIX B: DEFINITIONS	95
APPENDIX C: DATA BASE OF ORGANIZATIONAL CHARTS	97
APPENDIX D: QUESTIONNAIRE	119

ORGANIZING FOR EFFECTIVE SOFTWARE DEVELOPMENT

LIST OF EXHIBITS

			<u>Page</u>
II	-1	Rating Of The Overall Effectiveness Of Software Development Organizations By Group	7
III	-1	Overall Effectiveness Ratings Of Software Development Groups	19
	-2	Revenue, Profit, And Research And Development Data On Public Companies In Group I Compared To Public Companies In Groups II, III And IV	24
	-3	Individual Performances Of Public Companies From Group I	25
	-4	Individual Performance Of Public Companies From Groups II, III And IV	26
	-5	Weighted Average Productivity Of Respondent Public Companies	29
	-6	Categorization Of Vendors' Revenues And Employees By Group	31
	-7	Categorization Of Vendors	33
IV	-1	Level And Type Of Management To Which Software Development Department Reports	36
	-2	Groups By Organizational Type	38
	-3	Organizational Type By Group	40
	-4	Organizational Strengths And Weaknesses Self-Assessed By Project Organizations	41
	-5	Organizational Strengths And Weaknesses Self-Assessed By Functional Organizations	44
	-6	Organizational Strengths And Weaknesses Self-Assessed By Matrix Organizations	46
	-7	Number Of Companies Responding To The Question, "Are Maintenance And Development Of Software Separated Functionally?"	49
	-8	Development And Maintenance (Sustaining) Personnel By Analyst And Programmer Categories By Company Within Groups	52
	-9	Respondent Self-Ratings Of All Communications For All Companies By Groups	54

		<u>Page</u>
	-10 Quality And Effectiveness Of Software Development's Communications With Other Departments - Respondent Self-Ratings	55
	-11 Surveyed Companies' Suggestions For Interdepartmental Communications Improvements By Groups	57
V	-1 Selected Financial Data On Individual Companies From Group I	62
	-2 Selected Financial Data On Individual Companies From Group II	63
	-3 Selected Financial Data On Individual Companies From Group III	64
	-4 Selected Financial Data On Individual Companies From Group IV	65
	-5 Selected Financial Data On Individual Public Companies From Groups II, III And IV	66
	-6 Percent Of Responses Made By Groups To Questions On Selected Financial Data	68
	-7 Percent Of Total Software Budget Expended On Maintenance Function	69
VI	-1 Use, Effectiveness And Adherence To Standards	73
	-2 Responses To The Question, "If You Have Standards, How Do You Measure The Level Of Adherence To Those Standards?"	75
	-3 Level Of Quality Control	76
	-4 Responses To The Question, "How Do You Measure The Quality Of Your Software Development?"	78
	-5 Responses To The Question, "What Type Of Quality Control Mechanism Do You Employ?"	80
	-6 Responses To The Question, "What Procedural Improvements Would You Recommend For Increasing Software Product Quality?"	82
	-7 Responses To The Question, "How Do You Measure Software Development Success And Failure?"	84
	-8 Reasons For Software Development Successes And Failures In Past Five Years	86
VII	-1 Software Development Methodology Respondents Envision Being Available By 1985	90
C	-1 Company A	98
	-2 Company B	99
	-3 Company C	100
	-4 Company D	101
	-5 Company E	102
	-6 Company F	103
	-7 Company G	104
	-8 Company H	105
	-9 Company I	106

	<u>Page</u>
-10 Company J	107
-11 Company K	108
-12 Company L	109
-13 Company M	110
-14 Company N	111
-15 Company O	112
-16 Company P	113
-17 Company Q	114
-18 Company R	115
-19 Company S	116
-20 Company T	117
-21 Company U	118



Digitized by the Internet Archive
in 2015

<https://archive.org/details/organizingforeffunse>

I INTRODUCTION

I INTRODUCTION

- This report was prepared by INPUT as a custom study for GTE Service Corporation, Stamford, Connecticut.
- The primary objective of the study was to describe the software development procedures of computer manufacturing corporations who produce small- and medium-sized freestanding systems.
- Information for this study was obtained from several sources:
 - On-site interviews with eight computer manufacturers.
 - Telephone interviews with 13 computer manufacturers.
 - Review of INPUT's files and previous studies on the subject of software development and maintenance.
- The 21 companies interviewed had annual revenues of approximately \$24 billion, of which about \$12 billion was derived from the sale of computer systems and software. A profile of the companies interviewed and their organizational charts are in Appendices A and C.
- These companies represent almost half of the computer manufacturers in the United States.

- Many of the interviewees had total responsibility for the development of software for their company's computer products while others were responsible for a major division or product line. Titles of the people interviewed included:
 - Section Manager, General Systems Division.
 - Director, Advanced Systems Development.
 - Manager, Language Processor Group.
 - Manager, Technical Release.
 - Director of Network Engineering and Software Development.
 - Vice President, Software.
 - Manager, Software Engineering, Advanced Products.
 - Engineering Manager (Software).
 - Manager, Systems Engineering.
 - President.
 - Director of Software.
 - Manager, Systems Software Development.
 - General Manager, Division.
 - Vice President, Product Development (Software).
 - Industry Systems Software Manager.
 - Product Engineering (Software) Manager.
 - Director of Programming.
 - Director of Software.
 - Manager, Basic Software.
 - Director of Software.
 - Director, Software Products.

- None of the managers interviewed was responsible for in-house data processing or Management Information Systems.

- Although the companies interviewed represent a large percentage of the computer manufacturing industry, the data reported on in this study cannot be considered statistically reliable due the small sample of companies interviewed and, for many specific questions, the even smaller amount of data obtained.

- Nevertheless, the information and analysis is considered very useful in that it does give strong indications in many cases about factors affecting the development and maintenance of computer software.

II EXECUTIVE SUMMARY

II EXECUTIVE SUMMARY

A. MAJOR FINDINGS AND CONCLUSIONS

I. EFFECTIVE SOFTWARE ORGANIZATIONS

- The software development organizations of 21 companies were interviewed, analyzed and evaluated.
- A methodology was developed for evaluating the software development organization's effectiveness and productivity based on an assessment of the software's market acceptance, profitability, quality, reliability and maintenance cost.
- The companies were rated on a scale of 1 to 4, where 1 was the least effective organization.
 - All of the companies that received the same rating were assigned to the same group, which was numbered according to the rating received.
 - The companies in Group I clearly out-performed the companies in Groups II, III and IV by every criteria of success applied to the company and its software development organization.

- The number of companies in each group and a description of the rating of each group are shown in Exhibit II-1.
2. SOFTWARE DEVELOPMENT AND CORPORATE ORGANIZATION
- a. Reporting Structure
- Seventy-five percent of the directors of software development reported to second level management, while 25% reported to the first levels.
 - A majority of the directors reported to the research and development vice president, and the balance reported to either the division manager (or president) or the marketing vice president.
 - More than sixty percent of the companies had a project-oriented organization, with the remainder evenly divided between function and matrix organizations.
 - The above-average companies favored project orientation first and matrix second; none was functionally organized.
 - Many of the companies had separate software groups dedicated to advanced systems development.
 - Some of the companies had internal consultants who acted as advisors on software technology to senior management.
 - A substantial difference was found between the groups on the issue of separate maintenance functions.
 - Eighty-three percent of the Group I companies separated maintenance functionally.
 - Only 20% of the Group II, III and IV companies separated development from maintenance.

EXHIBIT II-1

RATING OF THE OVERALL EFFECTIVENESS OF SOFTWARE
DEVELOPMENT ORGANIZATIONS BY GROUP

GROUP	RATING	NUMBER OF COMPANIES
I	ABOVE AVERAGE	6
II	AVERAGE	6
III	BELOW AVERAGE	7
IV	WELL BELOW AVERAGE	2

- Data gathered on the number of analysts and programmers employed indicated that:
 - Most companies did not differentiate between the two job classifications.
 - Group I companies provided more detail than the other companies.

b. Interdepartmental Relationships

- The surveyed software departments were asked to rate the quantity and effectiveness of their communications with other departments.
- The most effective communications were between software maintenance and hardware development departments.
- The most important interfaces were with:
 - Product planning.
 - Market planning.
 - Business planning.
- The overall communications of the different groups were rated at about the same level for each group.

c. Organizational Changes

- Most of the companies surveyed were growing rapidly, and growth was the principal force causing organizational changes.
- The changes were usually towards more functional specialization or into new lines of business divisions.

3. COST AND FINANCIAL RATIOS

- Very little meaningful data were provided by the surveyed companies on software development cost.
- Software development was found to cost twice as much as hardware development.
- Maintenance is about 60% of the software budget.
- Most respondents had difficulty distinguishing between sustenance and enhancement of software.
- The most effective organizations had the best understanding of their costs.

4. PROGRAMMING STANDARDS

- Eighty-one percent of all the companies reported that they employed programming standards.
- There may not be a direct correlation between the use of programming standards and the effectiveness of software development.
- Those companies that gave themselves high ratings for effectiveness and level of adherence to programming standards were among the lower-rated companies in effective software development.

5. QUALITY CONTROL

- The level of quality control employed by each company in the software development process was evaluated.
 - A high level of quality control was associated with the Group I companies.

- A much lower level was found in the lower-rated groups.
- The most effective software development departments employed quality control techniques that were:
 - Quantitative.
 - Scientific.
 - Statistical.
 - Historical.
 - Predictive.
- The surveyed companies frequently mentioned that they could improve their software product quality by improving the product specifications or the specification process.
 - Improved quality assurance was also mentioned frequently.

6. MEASURES OF SOFTWARE DEVELOPMENT SUCCESS

- The companies indicated that they most often employed the following measures of software development successes and failures:
 - The product was developed on schedule.
 - Customers have accepted the product.
 - The product works.
 - The product was developed on budget.

- The most frequently cited reasons for software development success were:

- Well-conceived and complete specifications.
- A thorough and well-designed product.
- Good controls.

- Reasons for software development failure included:

- Being out of touch with the marketplace.
- Specifications that were:
 - . Poor.
 - . Impossible.
 - . Changing.
- Inadequate reviews and signoffs.

7. FUTURE OF SOFTWARE DEVELOPMENT

- Respondent companies expect that the following software development methodologies will be available by 1985:

- More programming tools.
- Automatic program generators.
- Built-in standards.
- Higher-level language.

- Computerized documentation.

B. RECOMMENDATIONS

I. ORGANIZING FOR EFFECTIVE SOFTWARE DEVELOPMENT

a. Reporting Structure

- Software development managers should report to the director of research and development. They should not report to division heads or marketing managers.
- A project-oriented organization is most highly recommended.
- Matrix organizations can be very effective under excellent management but they are riskier than project organizations because a change in management or management objectives can very severely impair their ability to perform productively.
- Business unit organizations have the same weakness as the matrix organizations, but a number of companies have successfully implemented this option and it should be given thorough consideration.
- Functional organizations appear to be safest because they are rarely very good or very bad, but they usually yield the most mediocre software development results.
- The use of advanced development departments in major product lines should be encouraged because they provide:
 - Alternate non-management career paths for software developers.
 - A foundation for new divisions.

- Non-management career paths providing for high-level advancement of software personnel should be implemented.
 - Advancing software personnel to high-level management consultant positions within the research and development organizations is likely to improve the software development process.
- INPUT recommends that the maintenance function be separated from the development function.
- Development organizations should be organized so that software and hardware development engineers work as closely together as possible.

b. Interdepartmental Relationships

- Provide for good communications between the software development department and both in software maintenance and hardware development groups.
- The most important interfaces with software development include:
 - Product planning.
 - Market planning.
 - Business planning.
- The most vital interface for effective software development is with the end user of the product.
 - Communications between software development personnel and the end user should be encouraged in every way possible.
 - Combining a close interface with the customer and planning process is very important to the success of development.

- To improve productivity and effectiveness in developing high-quality software, management should:
 - Impart a "sense of mission" (i.e., company mission and goals) to the developers.
 - Convey how important software development effort is to that mission.
 - Increase awareness of how important product development efforts are to sales and revenues - for the company and the product.

c. Organizational Changes

- As development efforts grow, a quality assurance department should be established that reports to the director of research and development and is separate from software and hardware engineering.
- Large organizations should also employ advanced development teams.
- Large software development departments should have their own documentation and publications people.
- As the software organization grows, formal planning functions should be established in the group that include personnel from:
 - Marketing.
 - Software engineering.
 - Hardware engineering.
 - Business planning.

- Selective rotation of personnel from applications to systems development should be considered.
- Rotating personnel from development to maintenance is not recommended because the skills, talents, and personality traits required for each function are different.

2. COST AND FINANCIAL RATIOS

- Strong management, financial and productivity measures and controls must be employed to attain effective software development.

3. PROGRAMMING STANDARDS

- A study of how standards affect software productivity in the organization should be given consideration because there are indications that the type of programming standards employed by a company may be counter-productive to effective software development.

4. QUALITY CONTROL

- A high-level of quality control is extremely important to effective software development.
- Quality control should be measured and related to revenues and profits.
- Goals, not just measurements, should be established for quality control.
- Control of the specification process is particularly important.

5. MEASURES OF SOFTWARE DEVELOPMENT SUCCESS

- Surprisingly, very few companies measured the success of their software products development by its return on investment, yet this was the most important measurement employed.

- In addition, success should be measured by:
 - Increased sales.
 - Number of installations.
 - Number of customer complaints.
- Management should motivate software development personnel by clearly defining how the success of their efforts and products are measured and periodically reviewing their achievements with them.
- The single most important reason for software development success is having good, close relations between end users and development personnel.
- Other important factors in software development success include having good controls, measures, reviews and sign-offs at all stages of development.

6. FUTURE OF SOFTWARE DEVELOPMENT

- INPUT recommends:
 - Converting to languages such as PASCAL and "C."
 - Incorporating more software into hardware.
 - Creating mechanized program generators.

III ANALYSIS OF SOFTWARE DEVELOPMENT EFFORT

III ANALYSIS OF SOFTWARE DEVELOPMENT EFFORT

A. EVALUATION TECHNIQUE

- An analysis of the data gathered in the interviews did not yield any single measurement for the overall effectiveness of the software development effort.
- Special attention was given to:
 - Lines of code produced per person month.
 - Cost per line of code.
- Neither measure was found to be useful in evaluating the software development effort, for the following reasons:
 - Only 11 of the 21 companies provided sufficient data to derive these figures.
 - The code was developed in a variety of languages.
 - There were no means of evaluating the quality of the code produced.
 - The figures given in most cases were rough estimates.

- The cost per line of code was effected by pay scales that differed substantially for companies located in different parts of the country.
- The cost per line varied from \$0.17 to \$14.67, due to a lack of standard definitions of the variables involved.
- One of the companies reported that it produced 16,666 lines of code per person month, which was improbable, if not impossible.
- The research staff concluded that the only useful measurement of overall effectiveness would have to be subjective.
- The two interviewers were asked to rate each of the companies they interviewed (on a scale of 1 to 4, where 1 was the highest rating and 4 was the lowest) based on their judgement of the overall effectiveness of the software development organization. Specific factors to be assessed included:
 - The productivity of the organization.
 - The market success and acceptance of the product.
 - The profitability of the product.
 - The quality and reliability of the product.
 - And the cost and level of maintenance required for the product.
 - Each interviewer then reviewed the ratings given by the other, and the ratings were found to be consistent.
- Exhibit III-1 shows that the ratings were fairly evenly distributed between the first three categories and between on-site and telephone interviews. Category IV was exceptional in that only 2 companies received that rating and both were telephone interviews.

EXHIBIT III-1

OVERALL EFFECTIVENESS RATINGS OF SOFTWARE DEVELOPMENT GROUPS

RATING	NUMBER OF COMPANIES RATED		
	ON-SITE INTERVIEWS	TELEPHONE INTERVIEWS	TOTAL INTERVIEWS
I	3	3	6
II	2	4	6
III	3	4	7
IV	0	2	2
TOTAL	8	13	21

- Some of the Exhibits do not include data on Group IV because insufficient data were obtained from them.
- These ratings can be interpreted as follows:
 - I Above average.
 - II Average.
 - III Below average.
 - IV Well below average.

B. METHODOLOGY OF TESTING RATINGS

- Since the ratings were in large part the subjective judgements of the consultant, several other tests were applied to the ratings to validate the methodology as well as the ratings themselves.
- All of the interview questionnaires were reviewed in detail for inconsistencies within each group.
 - A number of inconsistencies were found, but they were generally justifiable.
 - For example, several widely respected and very successful companies were given ratings of III.
 - One received this rating because the success of the firm was not related at all to the effectiveness of the software development group interviewed.

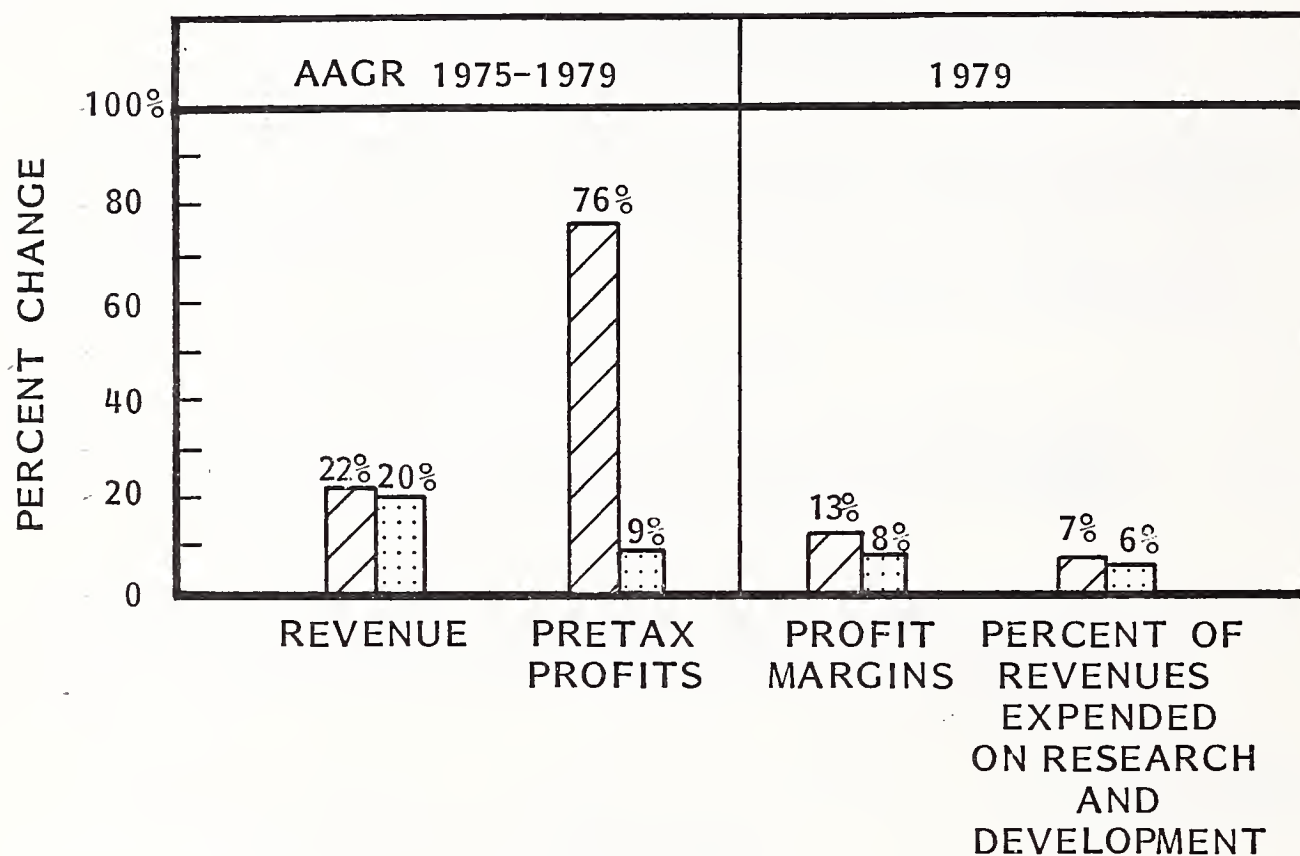
- . Another received this rating because there had been some major management changes in the past year that had had detrimental effects on the software development group.
- A number of apparent inconsistencies will be discussed later in the text of this report.
- The companies and their ratings were reviewed by several of INPUT's senior consultants who have extensive experience in the computer industry and a high level of knowledge of the companies interviewed.
 - The consultants generally agreed with the ratings.
 - In the few cases where the consultants did not agree with the ratings, they were unaware of extenuating circumstances such as those given above.
- A third test was applied to the ratings based entirely on verifiable empirical data.
 - This test was based on the hypothesis that there should be some correlation between the overall success of the corporation and the effectiveness of its software development group.
 - The success of the corporation would be gauged by several widely accepted measurements:
 - . Average annual growth rate in revenues for the past four years.
 - . Average annual growth rate in profits for the past four years.
 - . Pretax profit margins.

- . Expenditures on research and development as a percent of revenues.
- . Productivity as measured by revenue per employee.
- In order to qualify for this test, individual companies had to meet the following criteria:
 - . They had to be public companies.
 - . Computer manufacturing had to be a significant, if not pre-dominant, part of their business.
- Of the six companies rated I, five of them met this criteria.
- Of the remaining 15 companies in the three lower-rated groups, five met these criteria. They included:
 - . One II-rated company.
 - . Three III-rated companies.
 - . One IV-rated company.
- One of the companies had recently been acquired.
 - . Data from 1974-1978 were used for this company, rather than data from 1975-1979 as for the other companies.
 - . The data were judged to be consistent with the company's current performance and comparable to the data for other companies.

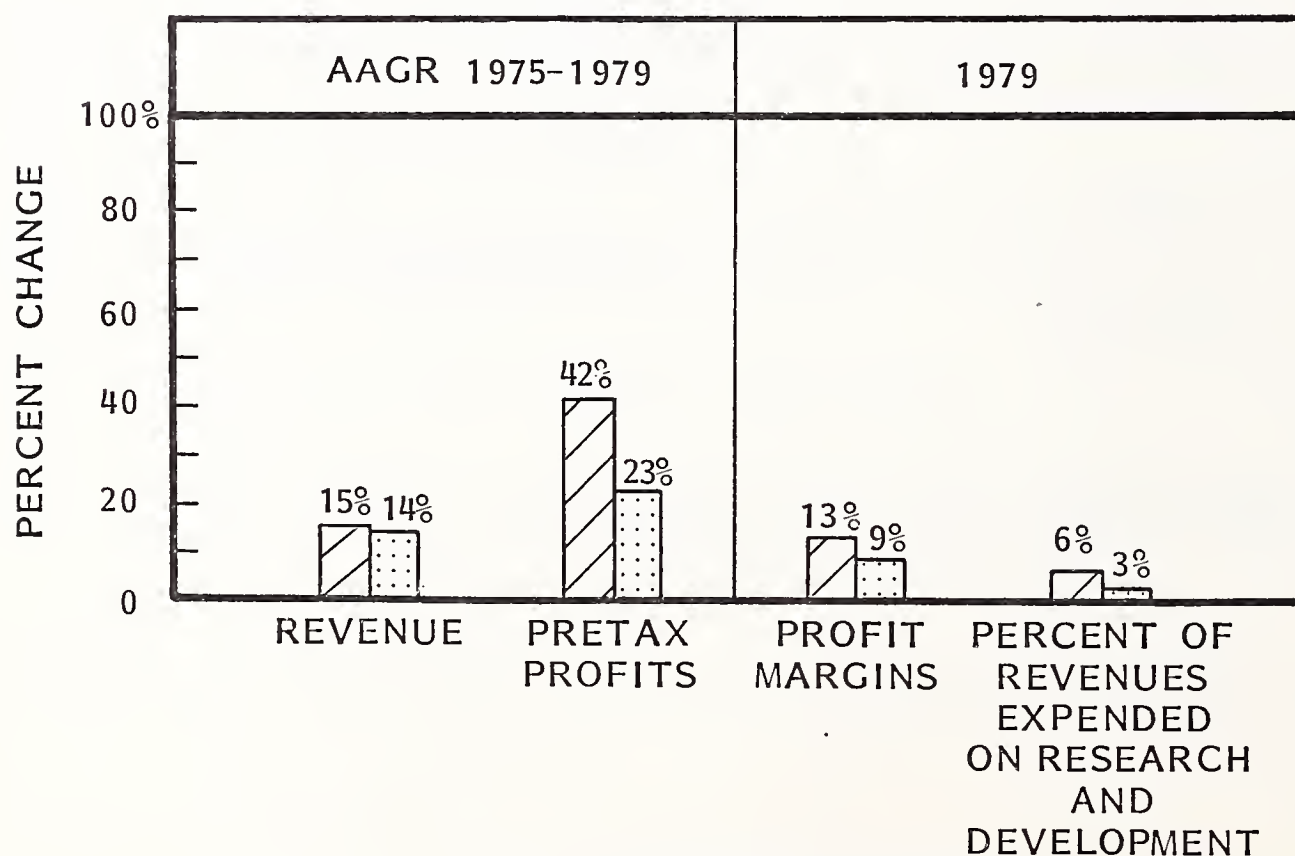
- One of the companies had an extremely volatile earnings record. Consequently, growth rates for a shorter period than 1975-1979 were used in the analysis, as they appeared to be a more accurate reflection of the company's performance.
- The revenue, profit, and research and development expenditures for the two sets of companies are shown in Exhibit III-2. The combined results were calculated in two ways:
 - The "average percentage" numbers were derived by averaging the individual percentages of each company in the set.
 - The "total percentage" numbers were derived by totaling the revenues of the individual companies in the set and calculating percentages based on total revenues.
 - The use of both techniques yielded essentially the same results.
- The public companies in group I clearly out-performed the public companies from groups II, III and IV in the revenue and profit performance measures.
- The highly rated companies also expended a substantially larger percent of their revenues on research and development than did the lower-rated companies.
- The sharpest distinction between the two sets of companies is seen in the pretax profit average annual growth rates.
- Comparable information for each of the five companies in both sets is shown in Exhibits III-3 and III-4.
 - Company C's 193% growth rate in pretax profits is due in part to very aggressive acquisition activity.

EXHIBIT III-2

REVENUE, PROFIT, AND RESEARCH AND DEVELOPMENT
DATA ON PUBLIC COMPANIES IN GROUP I COMPARED TO
PUBLIC COMPANIES IN GROUPS II, III AND IV



AVERAGE PERCENT CHANGE



TOTAL PERCENT CHANGE

-  GROUP I PUBLIC COMPANIES
-  GROUPS II, III, IV PUBLIC COMPANIES

EXHIBIT III-3

INDIVIDUAL PERFORMANCES OF PUBLIC COMPANIES FROM GROUP I

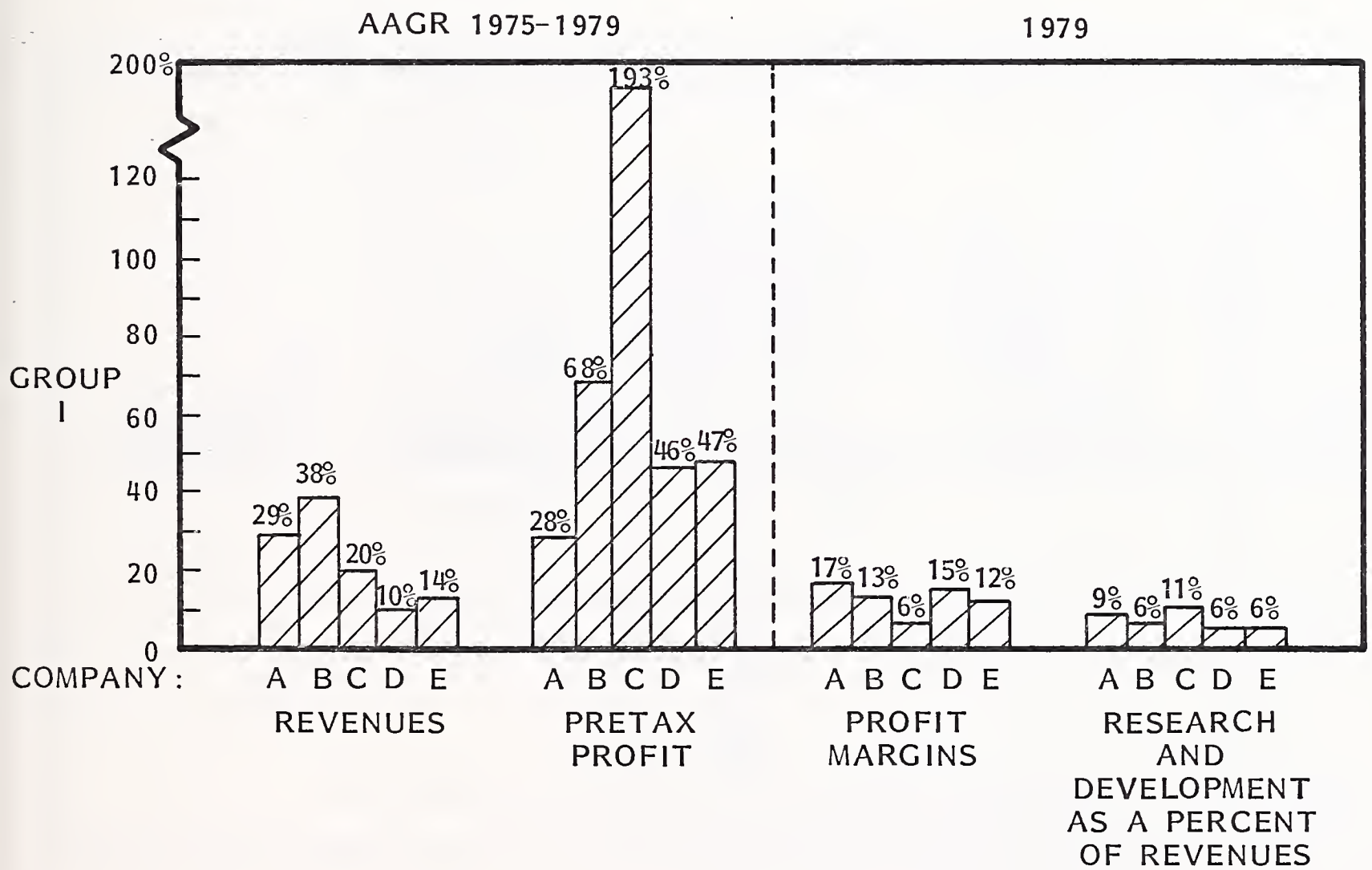
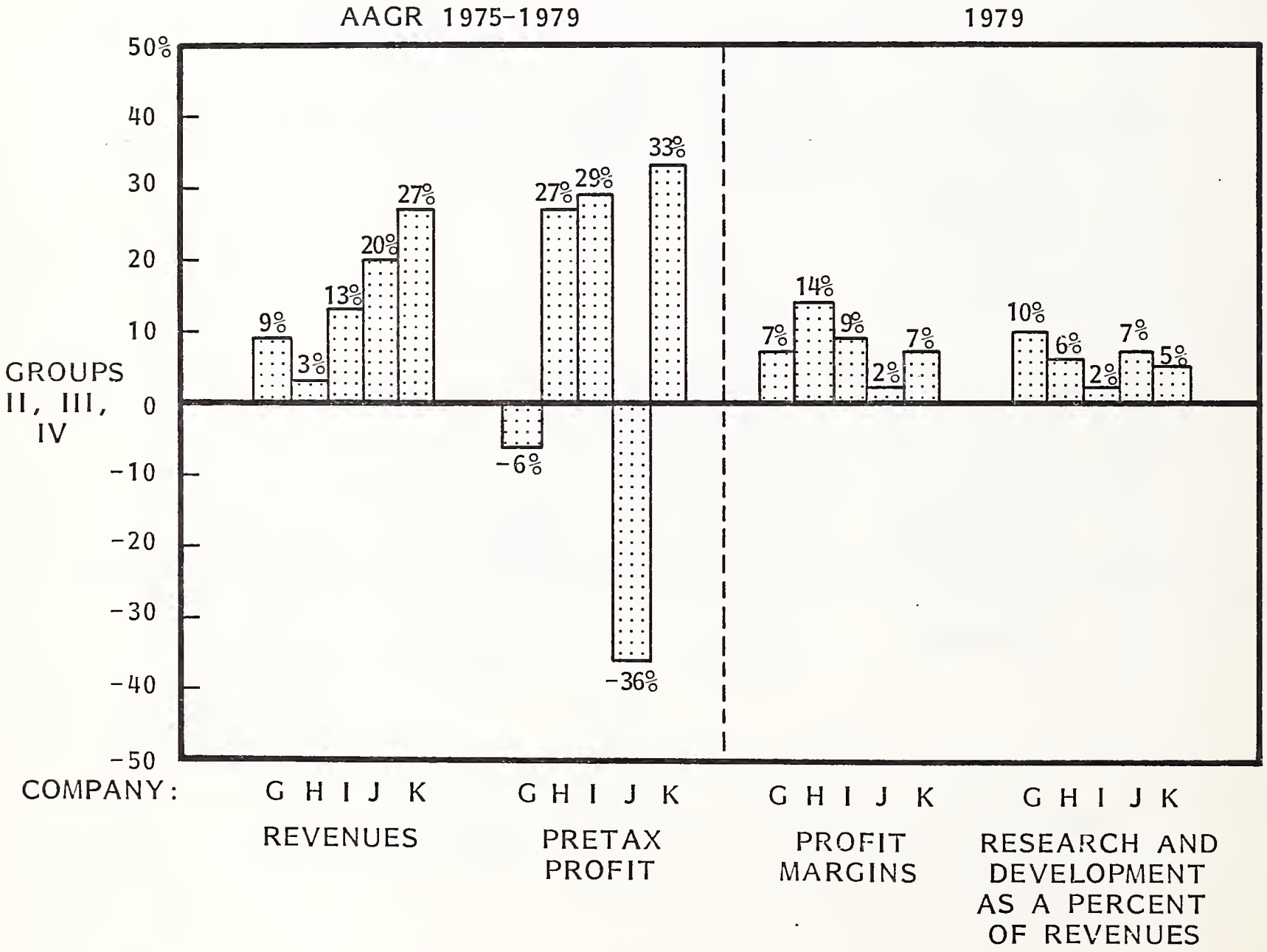


EXHIBIT III-4

INDIVIDUAL PERFORMANCE OF PUBLIC
COMPANIES FROM GROUPS II, III AND IV

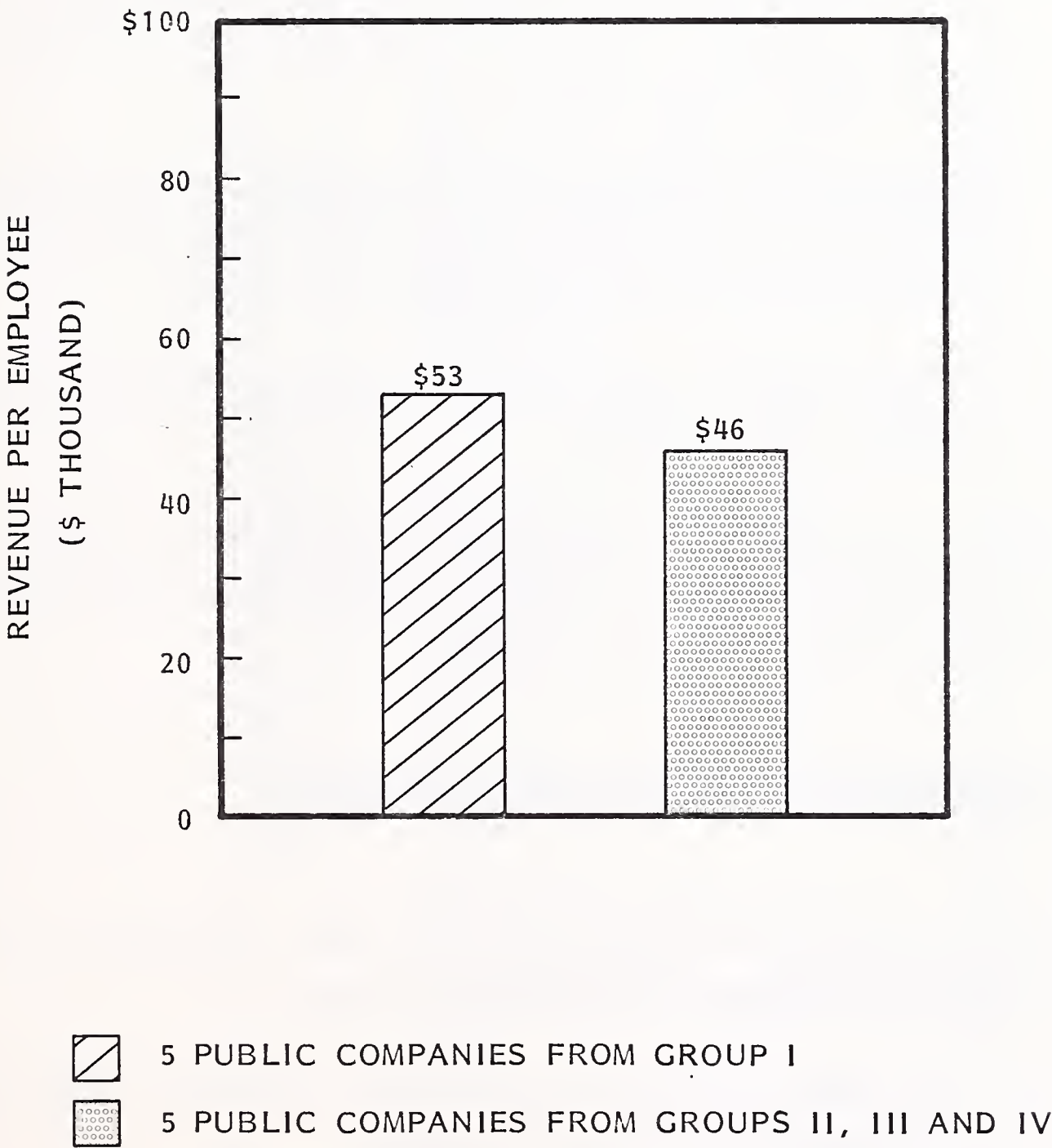


- Company K shows characteristics more compatible with the Group I companies than with its peers.
 - . This company received a III rating - even though it has high standards and produces quality products - because evidence of formal controls and procedures was not apparent.
 - . More importantly, a major new product was apparently being developed in a vacuum.
 - . There was very little contact between sales and customers, and very little concern for sales and profits, even by the division manager.
 - . After the ratings were assigned, the company announced a substantial drop in profits due to unexpectedly high research and development expenses for the new product.
 - . Management has had little previous experience with products of this type.
- The sixth company in Group I, the privately held company F, is known to have one of the most incredible revenue growth rates in the history of the computer industry.
 - It is believed to share the same financial characteristics as the other companies.
 - Although the company is substantially smaller than the other Group I companies, it is known for obtaining outstanding senior managers from other companies.
 - The company's director of software was responsible for developing the premier computer system of one of the other Group I companies.

- There is a substantial difference in size between the two sets of public companies.
 - Four Group I public companies have revenues over \$1 billion, and the fifth has \$200 million.
 - Groups II, III and IV public companies include one \$1 billion-plus company, one \$0.5 billion company and three companies with less than \$200 million in revenues.
- Generally, smaller companies are able to show greater percent growth rates than larger companies because the calculation is performed from a smaller base.
 - However, in this sample of public companies, the larger Group I far outperformed the smaller ones in Groups II, III and IV.
 - This unexpected phenomenon reinforces the qualitative differences between the two sets of companies.
- Exhibit III-5 shows productivity as measured by revenue per employee for the respondent public companies.
 - Productivity was 14% higher among Group I companies than it was among Groups II, III and IV companies.
- A fourth test of the ratings was employed based on users' ratings of software in a survey conducted by Datapro, a publisher of information about software.
- Five of the six vendors in Group I received ratings of E or Et, the two highest ratings given on a five-point scale.
 - The sixth vendor's software was not rated, but the manager of that company's software division was responsible, while employed at another firm, for the development of one of the systems rated Et.

EXHIBIT III-5

WEIGHTED AVERAGE PRODUCTIVITY
OF RESPONDENT PUBLIC COMPANIES



- One of the vendor's software products was also on Datapro's honor roll.
- None of the 15 companies in Groups II, III and IV offered software that was rated by Datapro's users.
 - The lack of a Datapro rating does not necessarily indicate poor-quality software.
 - Most of the software rated by Datapro is applications software, whereas much of the software the interviewees developed was systems software.
 - Many of the Groups II, III and IV companies did offer applications software but did not receive a rating. This implies that the software is either not widely used or not considered high quality.
 - It was surprising that none of the Groups II, III and IV companies' software was even rated.
- The conclusion drawn from this analysis of Datapro's user ratings is that it confirms INPUT's rating of the Group I companies and does not refute INPUT's rating of the Group II, III and IV companies.

C. CATEGORIZATION OF COMPANIES INTERVIEWED

- Characteristics of the four groups of companies interviewed in terms of revenues and employees are shown in Exhibit III-6.
 - The Group I company averages were substantially inflated by one company with over \$10 billion in revenues.

CATEGORIZATION OF VENDORS' REVENUES AND EMPLOYEES BY GROUP

GROUP	NUMBER OF COMPANIES	REVENUES 1979 (\$ MILLION)				EMPLOYEES			
		TOTAL	AVERAGE	HIGH	LOW	TOTAL	AVERAGE	HIGH	LOW
I	6	\$17,564	\$2,927	\$10,964	\$75	304,822	50,804	161,000	1,400
II	6	5,609	935	5,000	1	82,541	13,757	60,000	11
III	7	624	89	250	1	12,995	1,856	5,000	25
IV	2	98	49	98	0.2	2,020	1,010	2,000	20
ALL	21	23,895	1,138	10,964	0.2	402,378	19,161	161,000	11

- With the above exceptions, the companies in Groups I and II are fairly comparable.
 - The companies in Groups III and IV are substantially smaller than the companies in Groups I and II.
 - Company size seems to be a factor in that organizational effectiveness in developing software is primarily a product of management's experience and sophistication.
 - Good management is definitely important to good software development, and the larger companies generally had more of it.
- The organizations were categorized by themselves and INPUT by type of company, type of software, type of software development, group type and complete size orientation, as shown in Exhibit III-7.
 - The vendors' categorizations exceed 100% in all categories because many of the vendors fit into multiple categories.
 - INPUT's categorizations were based on the primary or principal category and allowed only one category per company.
 - Almost all of the companies were hardware manufacturers that developed systems software.
 - Four of the organizations developed communications software for the computer switching systems they manufactured. Five organizations also developed applications software.
 - All of the companies in Group I were hardware computer manufacturers.

EXHIBIT III-7
CATEGORIZATION OF VENDORS

CATEGORIES	VENDORS' SELF-CATEGORIZATION		INPUT CATEGORIZATION*				
	NUMBER OF COMPANIES	PERCENT OF COMPANIES	NUMBER OF COMPANIES				
			ALL COMPANIES	GROUP			
				I	II	III	IV
HARDWARE COMPUTER MANUFACTURER	19	90%	11	6	2	3	0
OEM INTEGRATOR	12	57	10	0	4	4	2
SYSTEM INTEGRATOR	4	19					
SOFTWARE VENDOR	5	26					
COMPUTER SERVICES	1	5					
SEMICONDUCTOR COMPONENTS	1	5					
<u>SOFTWARE CATEGORY</u>							
SYSTEMS	20	95	13	5	2	4	2
COMMUNICATIONS	12	57	3	1	1	1	0
APPLICATIONS	12	57	5	0	3	2	0
MAINTENANCE ONLY	9	43					
<u>SOFTWARE DEVELOPMENT GROUP TYPE</u>							
NEW STARTUP COMPANY	2	10	4	0	1	3	0
NEW DIVISION OR GROUP IN EXISTING COMPANY	9	43	6	2	0	3	1
NEW PRODUCT IN EXISTING COMPANY	10	48	11	4	5	1	1
<u>COMPUTER SIZE ORIENTATION</u>							
MICROPROCESSOR	1	5	3	0	1	2	0
MICROCOMPUTER	1	5	1	1	0	0	0
<\$10,000	1	5	0	0	0	0	0
<\$20,000	1	5	1	0	1	0	0
SMALL (\$20,000 TO \$100,000) MINICOMPUTER	14	67	10	2	3	3	2
MEDIUM (>\$100,000 BUT <IBM 370-158)	6	29	5	2	1	2	0
LARGE (IBM 370-158 OR LARGER)	2	10	1	1	0	0	0
COMMUNICATIONS	0	0	0	0	0	0	0

*ALLOWING ONLY ONE CATEGORY PER COMPANY

- The tighter and higher quality control measures generally employed by the companies in Group I are believed to be related to those employed in the manufacturing process.
- Several of the companies in Group I stated that they tried to use the same procedures in software development as they did in hardware development.
- Organizations that indicated "maintenance only" meant that they also maintained software in addition to developing it.
- Almost 75% of the organizations produced computers in the small to medium size range.
- It should be noted that none of the Group I companies was new.

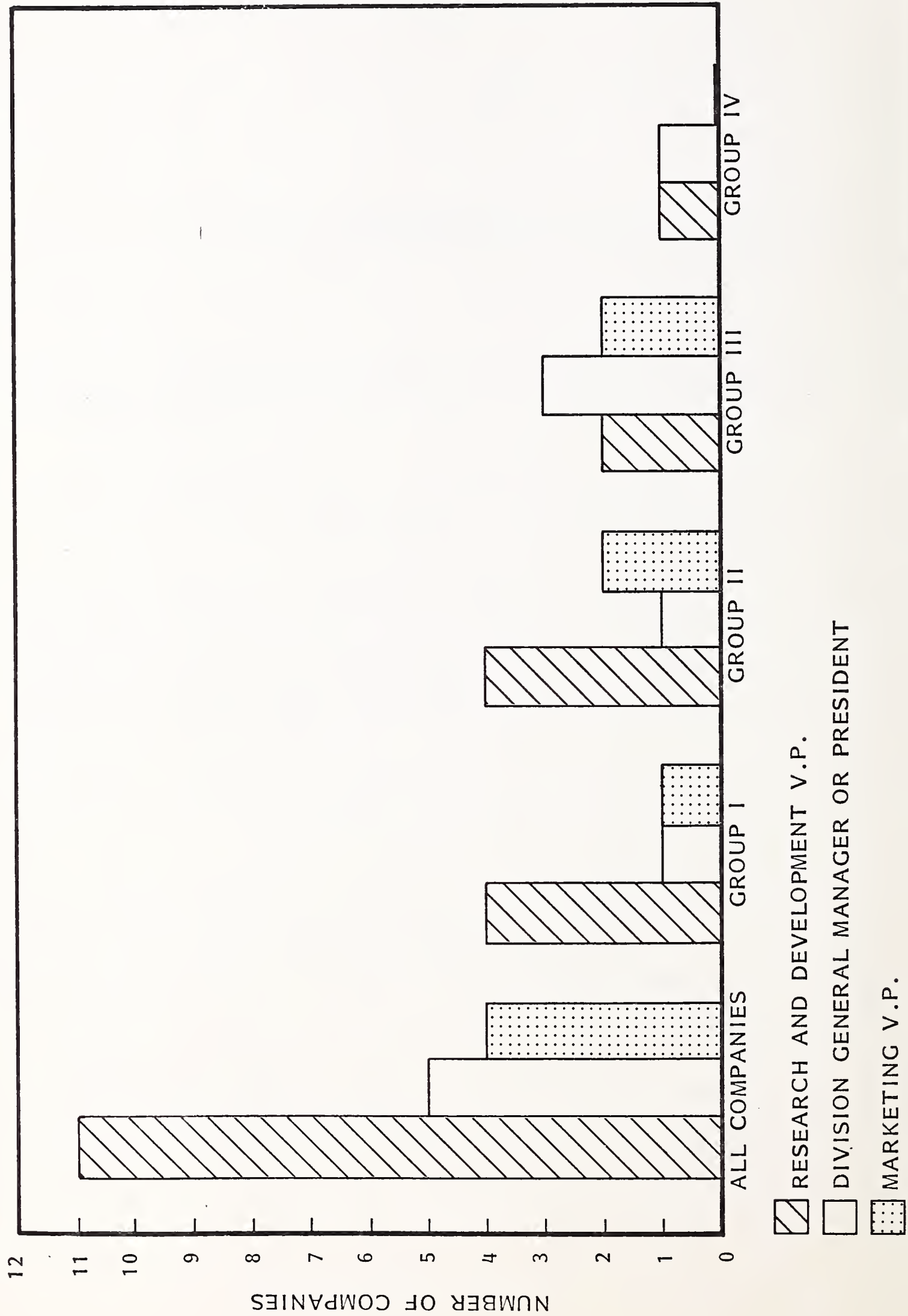
IV ORGANIZING FOR SOFTWARE DEVELOPMENT

IV ORGANIZING FOR SOFTWARE DEVELOPMENT

A. TYPICAL REPORTING STRUCTURES OF COMPANIES SURVEYED

- In all organizations, the manager of software development reported to high-level management, as shown in Exhibit IV-1.
 - The software manager reported to top-level management (the president or general manager of a division) in 25% of the organizations, and to a second-level manager in over 75% (eleven to the director of research and development and four to the vice president of marketing).
 - Those software managers that reported to top-level management were generally from smaller organizations.
 - Eighty percent of the cases where software reported to marketing were in Groups II and III, which indicates that reporting to marketing may not contribute to the effectiveness of the software development group.
- Reporting directly to top management also does not appear to be very beneficial to the software development department because the former probably does not have the time to be involved in the level of detail required to supervise the software group.

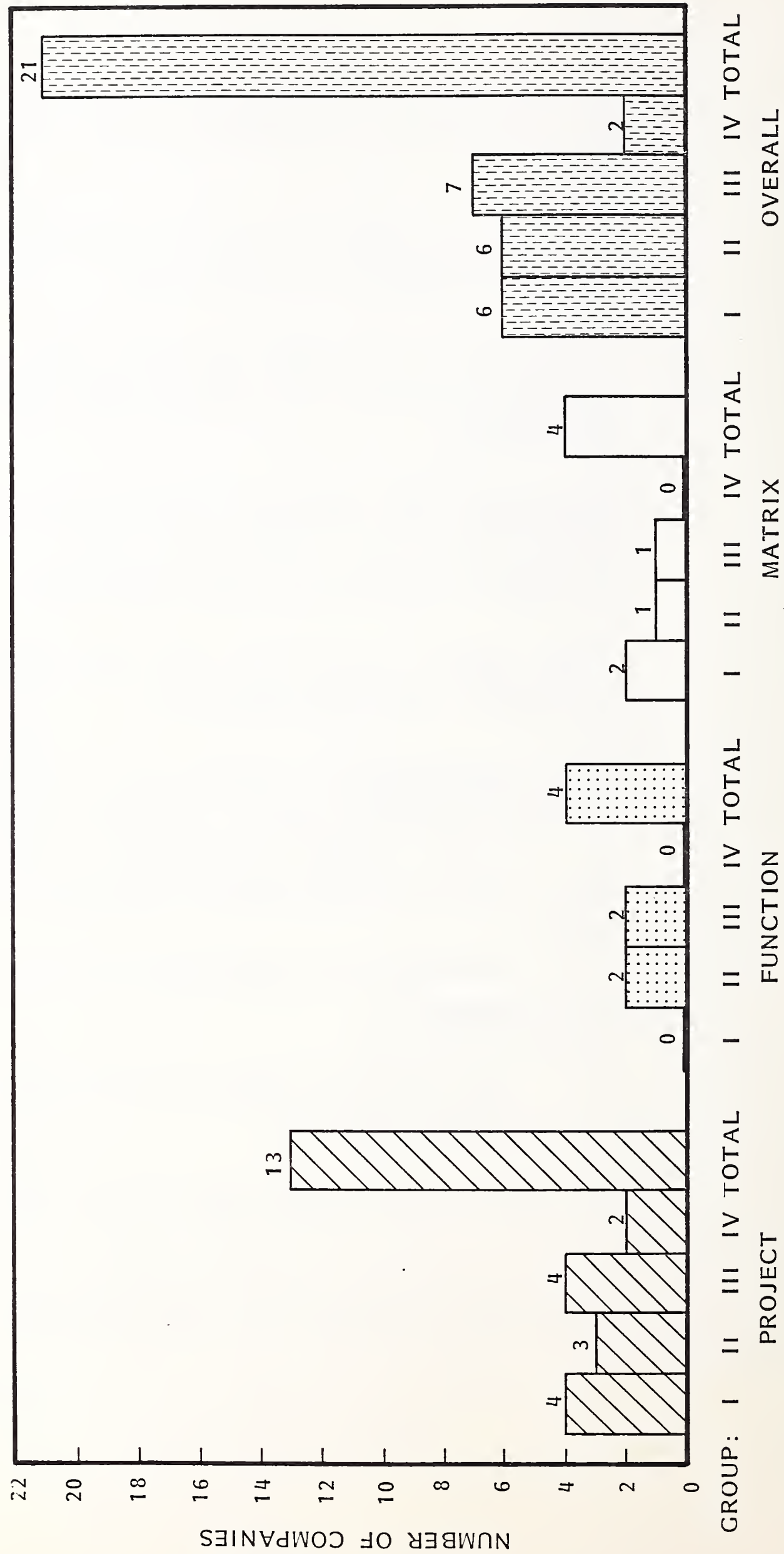
EXHIBIT IV-1
 LEVEL AND TYPE OF MANAGEMENT TO WHICH
 SOFTWARE DEVELOPMENT DEPARTMENT REPORTS



- All software development organizations surveyed were divided into three different types:
 - The Functional Type is organized so that individuals are assigned to specific functions that cross product lines.
 - The Project Type is organized so that individuals are assigned to specific products and may perform a variety of functions related to that product.
 - The Matrix Type is organized so that individuals share responsibility and/or authority for developing the software with other individuals outside of the software development organization.
 - One company that did not fit into these three categories had an organization based on the Business Unit.
 - The Business Unit contains all of the functions, or representatives of the functions, necessary to produce and market a product (including manufacturing, marketing, planning and quality assurance) assigned to a development group.
 - The business unit is supposed to have all of the capabilities of a company.
 - This company was categorized as a project organization for the purposes of this study.
- The distribution of groups by organization was analyzed as shown in Exhibit IV-2.
 - The majority of companies were organized by project.

EXHIBIT IV-2

GROUPS BY ORGANIZATIONAL TYPE



- The balance of companies were evenly divided between functional and matrix organization types.
- Group I companies had a larger percentage (67%) of project organizations than did the other groups, as shown in Exhibit IV-3.
 - None of the Group I companies had a functional organization.
 - One-third of the Group I companies had a matrix organization.
- INPUT believes that project organization contributes more to the success and effectiveness of the software development effort for the following reasons:
 - Software development people have a greater sense of identification with the product in this form of organization, which is a very important factor in the effective development of software.
 - This type of organization is most effective in imparting a sense of "mission" or purpose which is important to the development of good software.
 - Project management also requires smaller groups of people, which appear to be more effective in development than larger groups.
 - Responsibility and authority for the project are clearly defined and understood.
 - Frequent and good communication with the end user are very important to the software organization, and project management facilitates this more than the other types.
- The project organizations' comments on their own strengths and weaknesses appear in Exhibit IV-4.

EXHIBIT IV-3
ORGANIZATIONAL TYPE BY GROUP

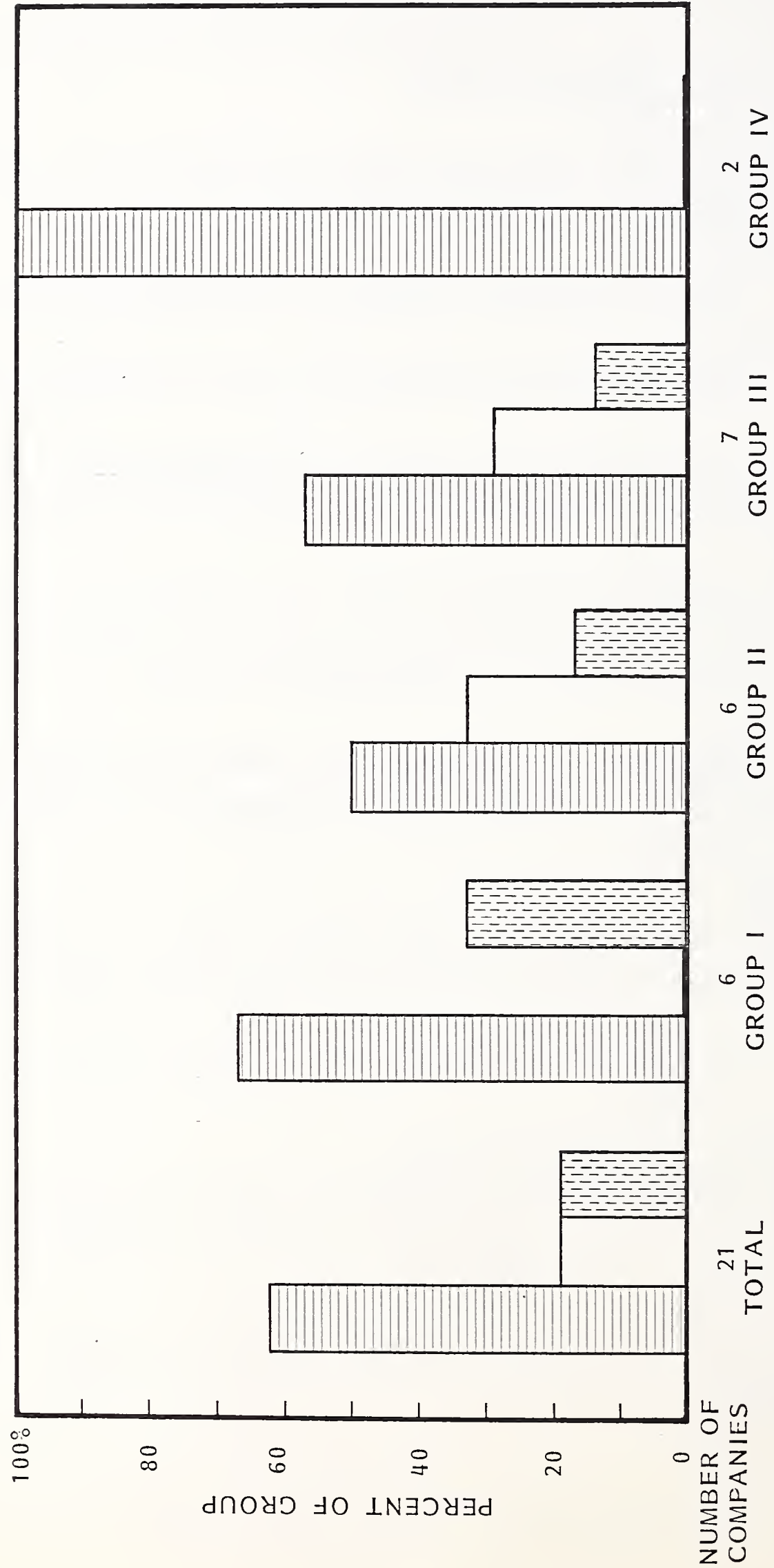


EXHIBIT IV-4

ORGANIZATIONAL STRENGTHS AND WEAKNESSES SELF-ASSESSED BY
PROJECT ORGANIZATIONS

GROUP I	STRENGTHS	WEAKNESSES
A	"Product identification is an ego motivator." "Good communications." "Manufacturing flexibility enables us to meet changing business needs."	"If strategy is not clearly defined, problems will arise." "Failure in objective setting process can cause changes in priorities."
B	"Care for the individual." "People are given freedom." "Good communications with the field organization."	"Lack of procedures."
C	"Small teams, two to four people." "We don't use 'chief programmer,' we have a project leader and everyone contributes." "We separate the development and sustaining functions, but rotate everyone from function to function."	"Too many critical people." "Separate custom programming group causes communications and cooperation problems."
E	"Our management approach has been to copy our hardware business in our software business." "Cost accountability."	"Inexperienced staff."
GROUP II		
G	"The small size of our department increases productivity and improves communications."	"We will require a more complex organizational structure if we expand."
N	"Ability to be responsive to the needs of our dealers and their customers."	
P	"Our chain of command is good and well defined." "Everyone has a strong sense of responsibility to each product, including testing and documentation."	"Limited growth for individuals, chances of becoming an analyst are not very good." "Sometimes too much concentration of knowledge in one person."

EXHIBIT IV-4 (CONT.)
ORGANIZATIONAL STRENGTHS AND WEAKNESSES SELF-ASSESSED BY
PROJECT ORGANIZATIONS

GROUP III	STRENGTHS	WEAKNESSES
I	"Separation of functions test and maintenance from the development group." "Designated systems designers." "Planned migration through organization for new college hires."	"Programmers identify so strongly with their product line that they resist moving to another product." "Planned migration also causes problems."
K	"Specialized products require functional divisions which provide strength of highly specialized people."	"We have problems with system integration."
Q	"Each individual in unit gets good overall view of products." "Diverse activities within unit provide challenging environment." "Excellent feedback from all operations produces pride of ownership and high motivation."	"Duplication of effort between units." "Very difficult to get help from other units."
S	None reported	None reported
GROUP IV		
J	"Informal organization."	"Organization is collapsing, we lost 80% of our software group in past two years."
U	"Good communications."	"Not enough depth in people due to small size."

- Functional organizations appear to be associated mostly with mediocre development efforts.
 - None of the top-rated or bottom-rated companies had this form of organization.
 - Functional organizations lack most of the benefits attributed above to project organizations.
- Functional organizations do provide several benefits towards software development.
 - They allow for more specialized skills that can be employed more efficiently.
 - They are safer in that fewer individuals become critical of a specific project.
 - They provide a more flexible environment for changes in assignments and priorities.
- The functional organizations' comments on their organizational strengths and weaknesses appear in Exhibit IV-5.
- Matrix organizations were developed to take advantage of all of the benefits of both project and functional organizations. This survey found that they work better in theory than in practice. Because of the complexity of this type of organization, more things generally go wrong than go right.
- Two of the Group I companies had a form of matrix management in their software development groups. In one the matrix was relatively simple:
 - Overall responsibility for the project was assigned to product managers.

EXHIBIT IV-5
ORGANIZATIONAL STRENGTHS AND WEAKNESSES SELF-ASSESSED BY
FUNCTIONAL ORGANIZATIONS

GROUP II	STRENGTHS	WEAKNESSES
M	"Specialized skills oriented towards tasks."	"No systems development in marketing."
O	"Separate support group which deals directly with dealers."	"No distinct research and development group. Research and development is performed by software development and manufacturing engineering, which are separate groups."
GROUP III		
R	"Each group does its own maintenance and development." "Chief programmers who have lots of responsibility including product specialization."	
T	None reported	"People are not assigned to a particular project."

- The mission was clear.
- Ample resources were available to do the job.
- The other Group I matrix organization, just emerging out of a project organization, was already developing problems:
 - The successful coordination of forces in the matrix were in the hands of a very high-level manager who appeared to be losing his grip due to the size and growth of the organization.
 - Conflicts were increasing between various managers that required frequent high-level management intervention.
 - Priorities were being disrupted or altered by marketing or development managers. The matrix organizations' comments on their organizational strengths and weaknesses appear in Exhibit IV-6.
- The business unit type of organization offers several advantages.
 - It is very close to a project type organization, especially in respect to identification with the product and communications with the end user.
 - The business unit manager has a great deal of authority along with full responsibility for the success of the product.
 - It does not require superb management and is not as subject to political forces as matrix management.
- The business unit company in this survey was rated in Group III due to very serious conflicts among the highest levels of management (resulting in some senior managers leaving the company), which disrupted the business unit's sense of mission and goals.

EXHIBIT IV-6
ORGANIZATIONAL STRENGTHS AND WEAKNESSES SELF-ASSESSED BY
MATRIX ORGANIZATIONS

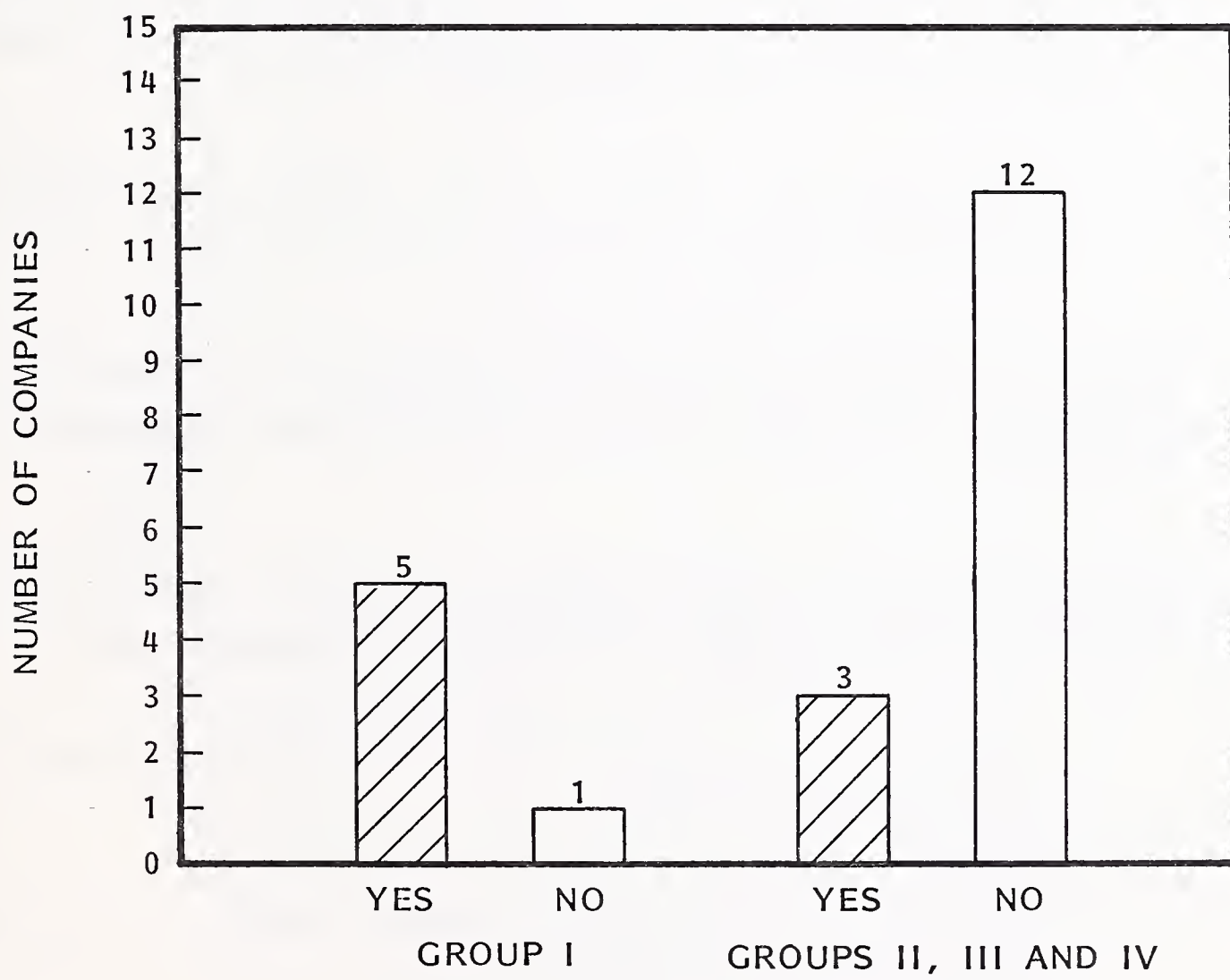
GROUP I	STRENGTHS	WEAKNESSES
B	"Each software group has its own quality assurance and publications group." "Great deal of pride and motivation to produce good products." "Standard development procedures produce high-quality software."	"Very large organization tends to lack control." "Lack of written communications causes losses in development time."
F	"I coordinate the development activities well." "Software development is a profit and loss center." "Each group will have its own engineering and publication group as we grow."	"Conflicts between marketing and development product managers."
GROUP II		
L	"Fully exploits specialized skills." "Provides uniform performance with no extreme failures or successes."	"No one person is responsible or accountable for a specific product." "People feel no identification with, or ownership of, product." "Decisions must be made through negotiations."
GROUP III		
H	"Broad range of technical challenges which motivate existing personnel and attract new hires." "In-depth industry and application expertise."	"Changes in priorities consume great deal of management time." "Many decisions must be made at high level in organization."

- The company's past record in software and hardware development is one of the best in the industry.
- It is a tribute to the business unit organization that everything hadn't fallen apart under the conditions this company experienced.
- A closer look at more organizations using the business unit concept would probably yield some very useful information.
- Many of the organizations, regardless of organization type, had a separate development department dedicated to advanced systems development.
 - This group was project-oriented.
 - It usually consisted of hardware as well as software engineers.
- The Advanced Development Department was used in several ways by the company's survey:
 - When the product development was completed, the nucleus of people in the department could be promoted to section heads in a new division which would continue development for the product through its life cycle.
 - In other companies, the department would then move on to developing the next future product as each one was completed.
 - It provided an alternative high level career path for the individual.
- A number of companies provided career advancement of up to five levels without imposing management responsibility on key creative software personnel.

- In one Group I company, the head of the department stated that he had a number of people working for him who were higher paid than he was.
- Some companies had positions known as "consultants" who reported to the top management in the software organizations.
 - These individuals were often considered geniuses or wizards.
 - They usually had no specific standing assignments.
 - They were often employed to keep the head of software in touch with the most advanced technology.
 - The position offered another career path for the nonmanagement-oriented software engineer.
- There was a striking difference between the Group I companies and all the other groups on the issue of separate maintenance functions, as shown in Exhibit IV-7.
 - Eighty-three percent of the Group I companies separated maintenance functionally.
 - Only 20% of the groups II, III and IV companies separated development from maintenance.
- The one respondent in Group I that said maintenance was not a separate function was part of a relatively new and small department.
 - That interview also stated that maintenance was a separate function in some of the larger departments in the company.

EXHIBIT IV-7

NUMBER OF COMPANIES RESPONDING TO THE QUESTION,
"ARE MAINTENANCE AND DEVELOPMENT
OF SOFTWARE SEPARATED FUNCTIONALLY?"



- The separation of maintenance from development did seem to be partly a function of size and/or expected longevity of the software development group's product.
- Many of the companies that did not separate maintenance from development felt very strongly that it was beneficial to combine the functions for the following reasons:
 - Maintenance was often considered to be a lower-level or less desirable job than development, and it was felt to be better for morale if everyone shared in this unpleasant task.
 - Some felt that the developer was best qualified and most efficient to maintain the product.
- One of the Group I companies addressed the issue of lower job status for maintenance by rotating all personnel from maintenance to development and back again.
- The separation of the two functions was overwhelmingly favored by the more successful companies, and is probably a factor in their success, because:
 - A development group that does not have maintenance responsibility is less subject to interruptions.
 - The skills, talents and personality traits are different for good maintenance people and good development people.
 - The activities are sufficiently different to benefit from different management, control and measurement techniques.
 - The separation of the two functions requires much better specifications and documentation.

- Maintenance personnel have a broader base of experience in dealing with a variety of problems, especially when a new product is released, because they will have seen similar problems in older products.
- All of the companies surveyed were asked to separate their development and maintenance personnel by analyst or programmer function, and to report the number of people in each activity.
 - A very large percentage of the companies reported that they did not distinguish between the functions of analyst and programmers because most of their people did both.
 - Only 3 of the 21 companies (14%) could report on the functions separately, as shown in Exhibit IV-8.
- Several of the companies, especially those engaged in the development of microprocessors, make very little distinction between software and hardware engineers.
 - So much software is being developed in hardware to yield firmware that development personnel have to be qualified in both hardware and software development.
 - In one of the Group I companies, individuals were assigned hardware development positions who had no formal training in college in that function.
- Although the total number of people dedicated to maintenance for companies who reported this breakdown was greater than the number of people dedicated to development, the average company had more personnel assigned to development than to maintenance.
- A surprising 73% of the more than 4,000 software personnel could not be accounted for as either development, maintenance, analyst or programmer.

EXHIBIT IV-8

DEVELOPMENT AND MAINTENANCE (SUSTAINING) PERSONNEL BY
ANALYST AND PROGRAMMER CATEGORIES BY COMPANY WITHIN GROUPS

COMPANY	DEVELOPMENT			MAINTENANCE			DEVELOPMENT AND MAINTENANCE	
	ANALYST	PROGRAM-MER	TOTAL	ANALYST	PROGRAM-MER	TOTAL	DETAILED RESPONSE TOTAL	OVERALL RESPONSE TOTAL
GROUP I	A	-	-	-	-	-	-	30
	B	-	80	-	-	20	100	100
	C	-	-	-	-	-	-	50
	D	-	-	-	-	-	-	2,200
	E	180	360	270	270	540	900	900
	F	-	42	-	-	18	-	60
SUBTOTAL	180	180	482	270	270	578	1,060	3,340
GROUP II	G	-	-	-	-	-	-	10
	L	-	21	-	-	4	23	25
	M	4	7	3	5	8	15	15
	N	-	-	-	-	-	-	3
	O	-	-	-	-	-	-	9
	P	-	-	-	-	-	-	37
SUBTOTAL	3	4	28	3	5	12	40	99
GROUP III	H	-	-	-	-	-	-	35
	I	-	-	-	-	-	-	50
	K	14	20	0	0	0	20	20
	Q	-	7	-	-	8	13	15
	R	-	-	-	-	-	-	34
	S	-	-	-	-	-	-	4
	T	-	-	-	-	-	-	15
SUBTOTAL	6	14	27	0	0	8	35	173
GROUP IV	U	-	-	-	-	-	-	3
	V	-	-	-	-	-	-	15
SUBTOTAL	-	-	-	-	-	-	-	18
GRAND TOTAL	189	198	537	273	275	598	1,135	3,612

- INDICATES NO RESPONSE TO THE QUESTION

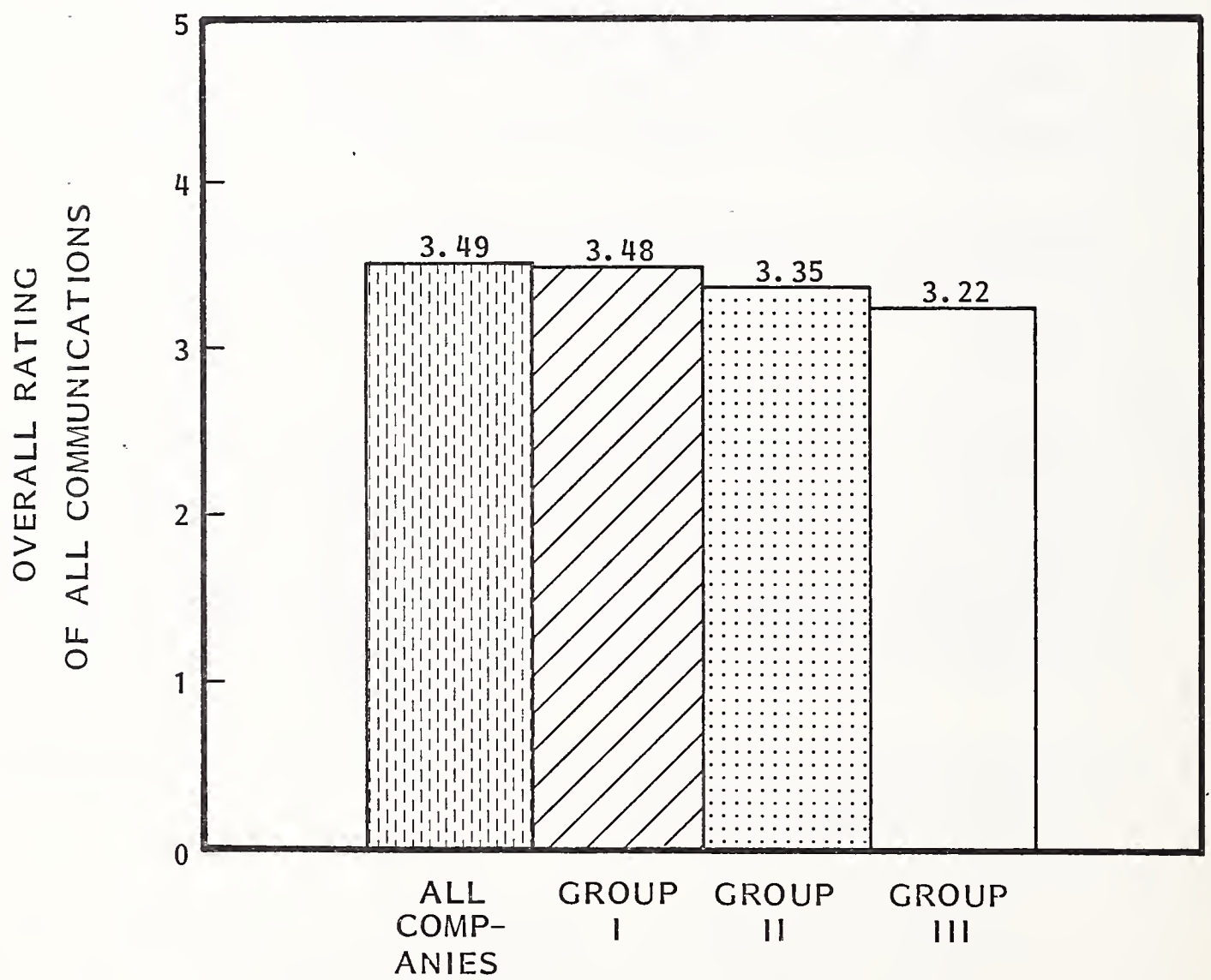
- The Group I companies were able to provide more detail on this subject than any of the other groups.

B. INTERDEPARTMENTAL RELATIONSHIPS

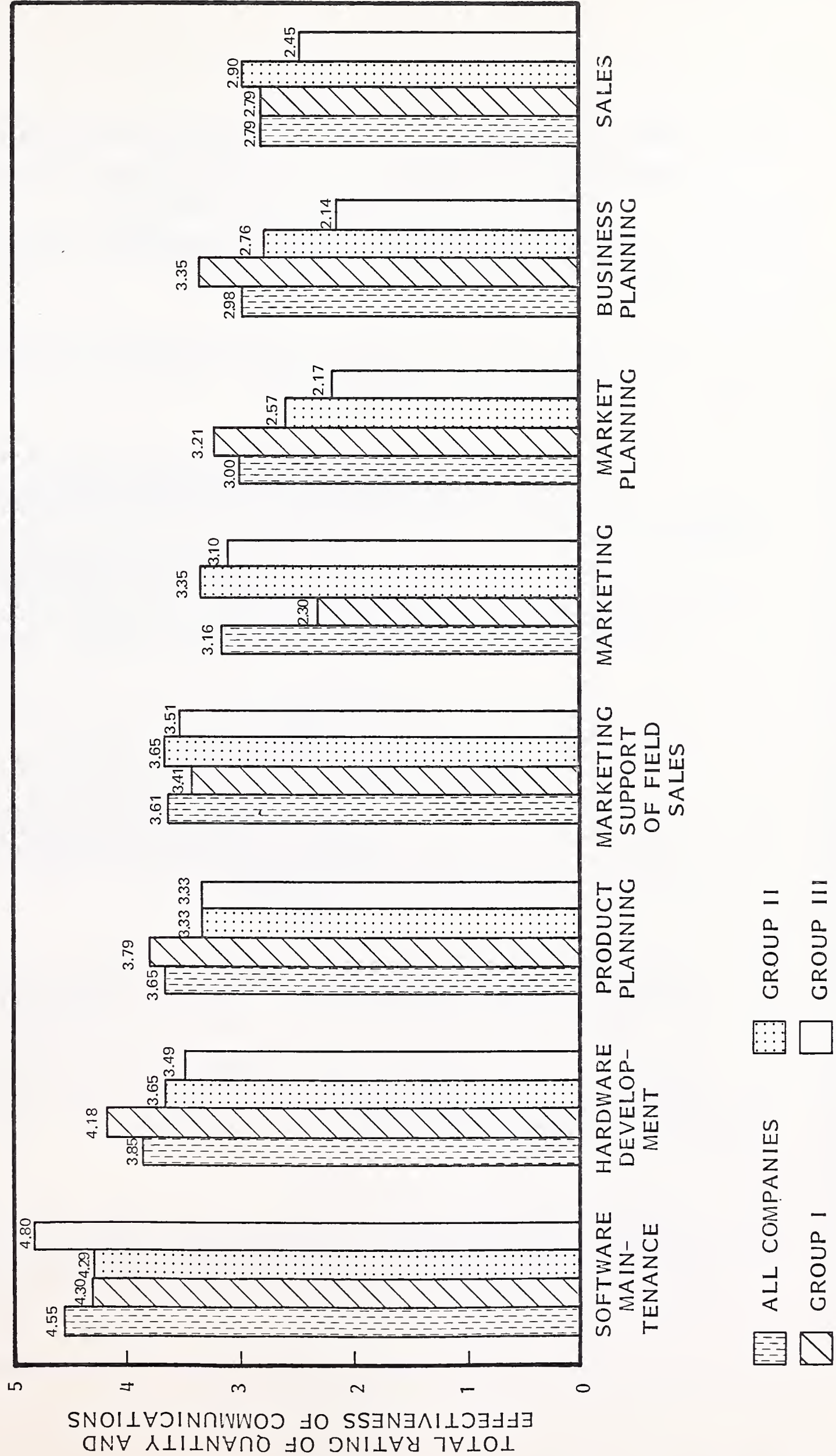
- The companies were asked to rate the quantity and effectiveness of communications between software development and other departments on a scale of 1 to 5, where 1 is the smallest quantity and least effective, and where 5 is the highest quantity and most effective. A total rating was then developed by multiplying the two ratings together and taking the square root of the results.
- Group IV consisted of only two companies, one of which had a total number of 20 employees who "all talked with each other," so it was not included in this analysis.
- Exhibit IV-9 indicates that the overall effectiveness of communications between the software development department and all other departments might be an important factor, since the higher-rated groups appear to communicate progressively better. However, the resultant ratings are so close that one cannot draw a firm conclusion from them.
- An examination of communications with eight different departments yields some more conclusive information.
 - The highest-level and most effective communications were with the software mainframe and hardware development for all three groups of software development organizations, as shown in Exhibit IV-10.
 - The interface with product, market and business planning was much more effective among Group I companies than among Group II and III companies.

EXHIBIT IV-9

RESPONDENT SELF-RATINGS OF ALL COMMUNICATIONS
FOR ALL COMPANIES BY GROUPS



QUALITY AND EFFECTIVENESS OF SOFTWARE DEVELOPMENT'S COMMUNICATIONS WITH OTHER DEPARTMENTS-
RESPONDENT SELF-RATINGS



- INPUT has observed that the effectiveness of the software development department is very dependent upon good communications with the end user.
- The communications with the end user were reinforced in the most successful organizations by:
 - Imparting to the software personnel the importance of sales and profits from the products they developed.
 - Establishing that the end users, and the sales force that serves them, are the software department's customers and responsibility.
 - Clearly defining the goals and mission of the company and the important role that software development had in attaining the goals and fulfilling the mission.
 - Emphasizing that budgets and schedules were very important, but not as important as a superior product and a happy customer.
- Communications with the planning departments are important not only in resolving marketing and technical problems, but in solving problems and providing needed solutions to the customer.
- The surveyed companies' suggestions for improving interdepartmental communications are shown in Exhibit IV-II.

C. ORGANIZATIONAL CHANGES

- Most of the companies surveyed were growing rapidly, and growth was the principal force causing organizational changes.

EXHIBIT IV-11

SURVEYED COMPANIES' SUGGESTIONS FOR INTERDEPARTMENTAL COMMUNICATIONS IMPROVEMENTS BY GROUPS

GROUP I

- "More communications in and before specification process."
- "Rotate people between marketing and programming."
- "Expose development people more to customer sites so they can see problem."
- "Establish procedures to resolve differences between marketing and software developing regarding priorities."
- "Need to communicate more in memos about problems in early stages before they become a big deal."
- "Communications between marketing and software development must be improved regarding objectives and priorities."

GROUP II

- "A recent re-organization has disrupted communications."
- "Status reports between groups should be more consistent."
- "More meetings and presentations."
- "Plan specific meetings to discuss issues."
- "We need more technical expertise at executive level."
- "Marketing is doing too much, going beyond their roles."

GROUP III

- "Follow procedures, write specifications and have formal signoffs."
- "Prior to commencement of a project, identify all the people that will be relevant to it and include them on an official list of who should receive information."
- "More frequent and regular meetings with published minutes of various development groups - its done only quarterly now."
- "Physical location is a problem. Marketing and software are in two different towns."

GROUP IV

- "We're only twenty people (entire company). We talk to each other all the time."
- "As we get bigger it will be a problem."
- "That relates to understanding mechanisms for handling common objectives."

- Many of the changes were toward more functional specialization as the organizations grew.
 - But there was a very clear effort, to maintain the project orientation of the software development department, which was more successful with the Group I companies.
 - Restructuring into divisions or lines of business was also utilized to maintain the project structure.
- As organizations grow, they often establish an advanced development team for new product development.
- The larger, more successful organizations often establish a separate, and sometimes very high level, quality assurance department.
- The creation of a publications and/or documentation group within each product group seems to contribute to the productivity of software.
- The organizational distance between hardware engineering and software engineering is becoming an important factor in developing good products.
 - The closer hardware and software engineering work together, the better the products.
- As the software organization gets larger, it becomes more important to have a product planning function in the group that contains:
 - Marketing personnel.
 - Software engineers.
 - Hardware engineers.

- Business planners.

- Whether large or small, companies appear to produce better software products when they do not report to marketing.
- Although most companies functionally separate systems from applications software development as they grow larger, some of the most successful companies maintain very close interfaces between these functions. Rotating personnel between these groups could be beneficial.
- Matrix organizations should only be attempted by very mature organizations with proven and stable management - and only as a last resort.

V SOFTWARE DEVELOPMENT COST AND
FINANCIAL RATIOS

V SOFTWARE DEVELOPMENT COST AND FINANCIAL RATIOS

- A number of questions about software development cost were asked in the interview, but very little meaningful information was provided by respondents.
- As mentioned earlier, the data provided on cost per line and lines per person month turned out to be of little use.
- Enough data were provided on software development and maintenance cost as a percent of hardware development cost to determine that it is approximately 200%, as shown in Exhibits V-1 through V-5.
- Respondents reported that the cost of software development was approximately three times as high as the cost of software maintenance, which conflicts with the relatively equal number of people assigned to those functions, as shown in Exhibit IV-8.
 - This difference is probably due to the fact that many of the development groups interviewed were involved more in new product development than in well-established products.
 - INPUT estimated that on established products, maintenance requires about 60% of the software budget.
- A number of the respondents did not know if they should classify enhancement to a product as maintenance or as development.

EXHIBIT V-1
SELECTED FINANCIAL DATA ON INDIVIDUAL COMPANIES FROM GROUP I

COMPANY*	SOFTWARE DEVELOPMENT COST AS A PERCENT OF REVENUE	SOFTWARE MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	LINES OF CODE PER PERSON MONTH	COST PER LINE OF CODE	TYPE OF CODE	TYPE OF COMPANY	EXPENDITURE FOR SOFTWARE DEVELOPMENT AND MAINTENANCE EMPLOYEE PER YEAR
A	-	-	0.5%	-	-	300%	209	\$14.34	SYSTEM & BASIC	HARDWARE	\$50,000
B	4.8%	1.2%	6.0	-	-	200	416	7.60	SYSTEM	HARDWARE	38,000
C	2.0	1.0	3.0	100%	100%	200	1,083	4.43	SYSTEM	HARDWARE	50,000
D	-	-	3.5	-	-	100	416	10.00	SYSTEM	HARDWARE	50,000
F	1.9	0.6	2.5	-	-	233	-	-	-	HARDWARE	45,000
	-	-	3.0	-	-	100	-	-	-	HARDWARE	50,000
AVERAGE	2.9	0.9	3.0	-	-	189	531	9.09			47,170

*PUBLIC COMPANIES ONLY ARE IDENTIFIED BY LETTER DESIGNATION

EXHIBIT V-2

SELECTED FINANCIAL DATA ON INDIVIDUAL COMPANIES FROM GROUP II

COMPANY*	SOFTWARE DEVELOPMENT COST AS A PERCENT OF REVENUE	SOFTWARE MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	LINES OF CODE PER PERSON MONTH	COST PER LINE OF CODE	TYPE OF CODE	TYPE OF COMPANY	EXPENDITURE FOR SOFTWARE DEVELOPMENT AND MAINTENANCE EMPLOYEE PER YEAR
G	-	-	-	-	-	-	-	-	-	-	\$60,000
	2%	.3%	2.3%	43%	10%	53%	687	\$8.48	-	HARDWARE	70,000
	-	-	-	-	-	-	16,666	.17	-	OEM	33,000
	-	-	5	-	-	200	3,333	1.00	-	OEM	30,000
	-	-	2	-	-	400	-	-	-	OEM	30,000
	-	-	7.5	-	-	NO HARDWARE	-	-	-	OEM	36,000
AVERAGE	2	.3	4.2		218		6,895	3.22			43,670

*PUBLIC COMPANIES ONLY ARE IDENTIFIED BY LETTER DESIGNATION

EXHIBIT V-3

SELECTED FINANCIAL DATA ON INDIVIDUAL COMPANIES FROM GROUP III

COMPANY	SOFTWARE DEVELOPMENT COST AS A PERCENT OF REVENUE	SOFTWARE MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	LINES OF CODE PER PERSON MONTH	COST PER LINE OF CODE	TYPE OF CODE	TYPE OF COMPANY	EXPENDITURE FOR SOFTWARE DEVELOPMENT AND MAINTENANCE EMPLOYEE PER YEAR
K	6%	-	6%	-	-	150%	972	\$ 4.25	SYSTEM	HARDWARE	\$60,000
H	-	-	-	-	-	-	-	-	-	OEM	60,000
	-	-	-	-	-	100	219	14.67	SYSTEM	HARDWARE	40,000
	1	2%	3	-	-	300	833	3.75	SYSTEM	OEM	35,000
	-	-	-	-	-	200	-	-	-	OEM	30,000
I	-	-	0.7	-	-	167	-	-	-	HARDWARE	30,000
	-	-	-	-	-	NO HARDWARE	-	-	-	OEM	-
AVERAGE	3.5	-	3.2	-	-	183	674	22.67	-	-	42,500

SELECTED FINANCIAL DATA ON INDIVIDUAL COMPANIES FROM GROUP IV

COMPANY	SOFTWARE DEVELOPMENT COST AS A PERCENT OF REVENUE	SOFTWARE MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	LINES OF CODE PER PERSON MONTH	COST PER LINE OF CODE	TYPE OF CODE	TYPE OF COMPANY	EXPENDITURE FOR SOFTWARE DEVELOPMENT AND MAINTENANCE EMPLOYEE PER YEAR
J	-	-	-	-	-	200%	-	-	-	OEM	-
AVER- AGE	-	-	-	-	-	-	-	-	-	OEM	-
AVER- AGE ALL GROUPS	295%	-1.02%	-3.46%	-	-	200	-	-	-	-	\$44,278

EXHIBIT V-5

SELECTED FINANCIAL DATA ON INDIVIDUAL PUBLIC COMPANIES FROM GROUPS II, III AND IV

COMPANY	SOFTWARE DEVELOPMENT COST AS A PERCENT OF REVENUE	SOFTWARE MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF REVENUE	SOFTWARE DEVELOPMENT COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	SOFTWARE DEVELOPMENT AND MAINTENANCE COST AS A PERCENT OF HARDWARE DEVELOPMENT COST	LINES OF CODE PER PERSON MONTH	COST PER LINE OF CODE	TYPE OF CODE	TYPE OF COMPANY	EXPENDITURE FOR SOFTWARE DEVELOPMENT AND MAINTENANCE EMPLOYEE PER YEAR
G	-	-	-	-	-	-	-	-	-	HARDWARE	\$60,000
H	-	-	-	-	-	-	-	-	-	OEM	62,000
I	-	-	0.7%	-	-	167%	-	-	-	HARDWARE	30,000
J	-	-	-	-	-	-	-	-	-	OEM	-
K	6%	0	6	-	-	150	972	\$4.25	SYSTEM	HARDWARE	60,000
AVERAGE	-	-	-	-	-	159%	-	-	-	-	\$53,000

- Other respondents found it difficult to distinguish between sustenance and enhancement.
- Still others indicated that the product development was a continuing process.
- It is suspected that some of the expenditures for software development and maintenance per employee per year were unburdened, while others were burdened, which jeopardizes the accuracy of the information.
 - Regional difference in pay scale also affected this number.
- Perhaps one of the most meaningful observations derived from these responses was the respondents' varying ability to answer questions on software development cost and productivity.
 - To derive the percent of questions that could be answered, the number of answers was divided by the number of questions on software development cost and productivity for each group.
 - The Group I companies were able to answer 63% of the questions as compared to 44% from all of the companies combined, as shown in Exhibit V-6.
 - It seems that the better one is informed about one's business, the more effectively one can run it.
 - The companies who rated lower on the scale of 1 to 4 were less concerned about the cost of doing business.
- Exhibit V-7 shows the individual respondents' answers to the question addressing percent of total software budget expended on the maintenance function.

EXHIBIT V-6

PERCENT OF RESPONSES MADE BY GROUPS
TO QUESTIONS ON SELECTED FINANCIAL DATA

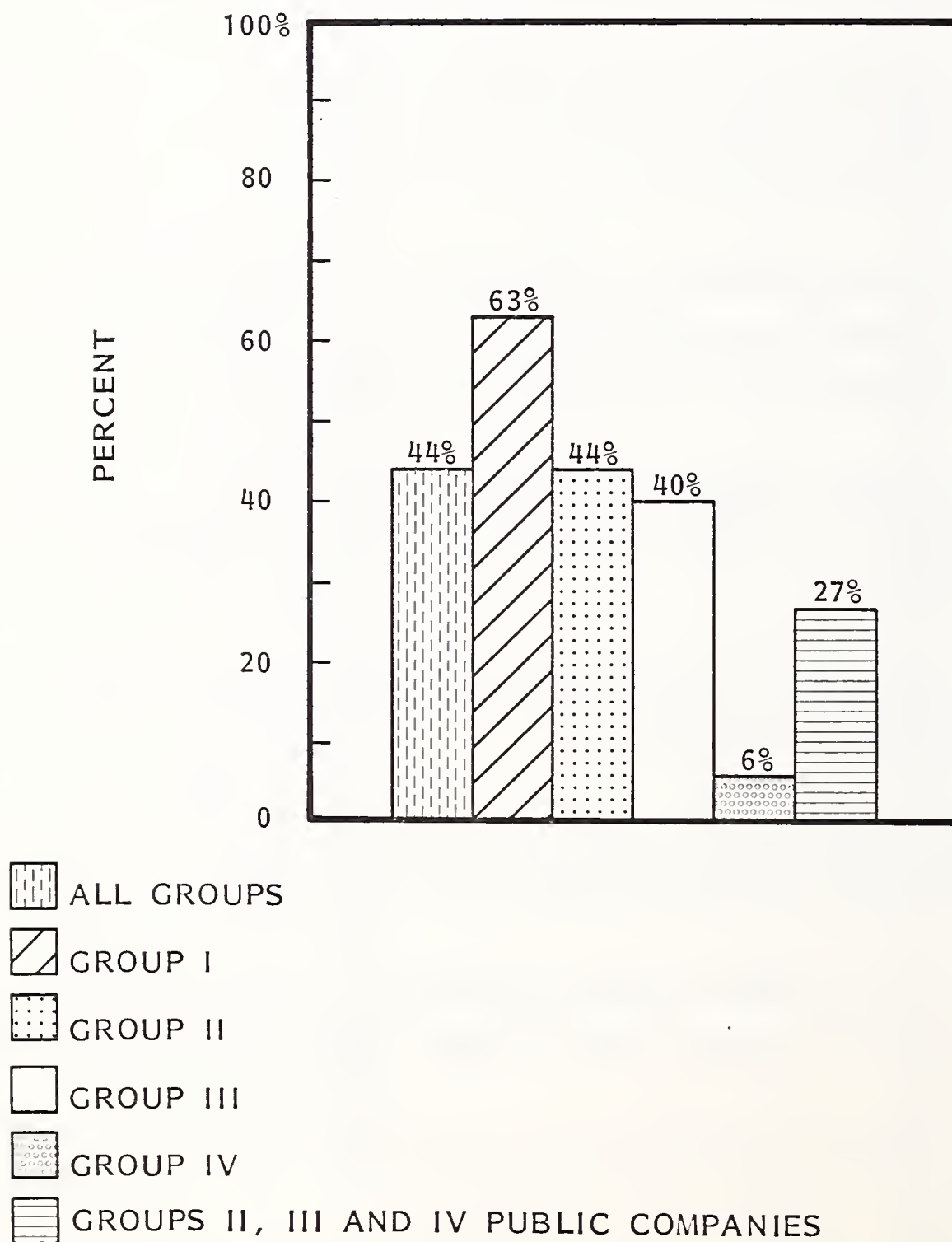
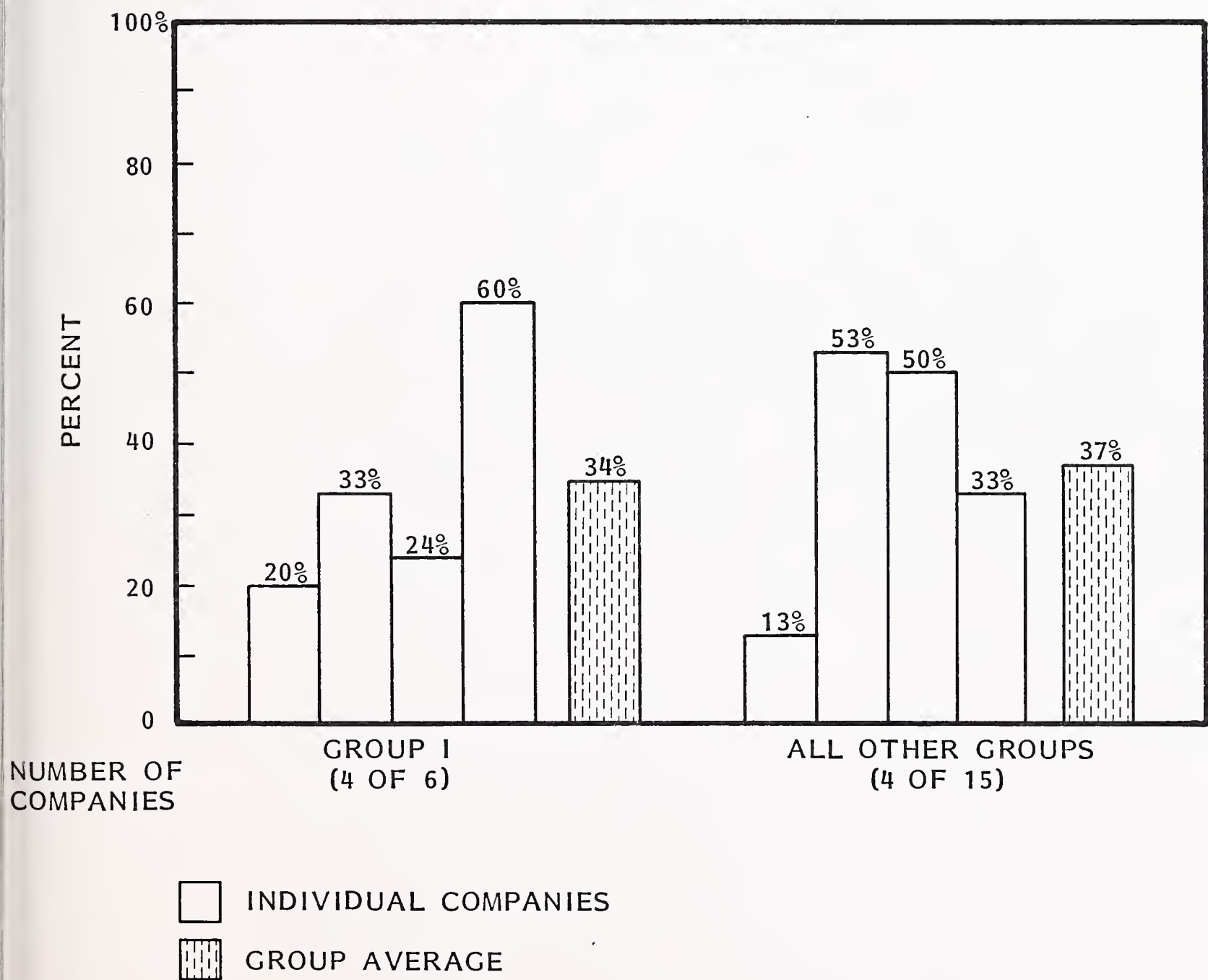


EXHIBIT V-7

PERCENT OF TOTAL SOFTWARE BUDGET
EXPENDED ON MAINTENANCE FUNCTION



- One would expect Group I to respond better to this question since a much higher percentage of these companies provided maintenance as a function separate from development.
- But even so, one would expect more than 27% of the Group II, III and IV respondents to be aware of maintenance cost.
- There was no significant difference in the amounts expended on the maintenance function between Group I and all other groups.

VI STANDARDS AND QUALITY CONTROL IN SOFTWARE DEVELOPMENT

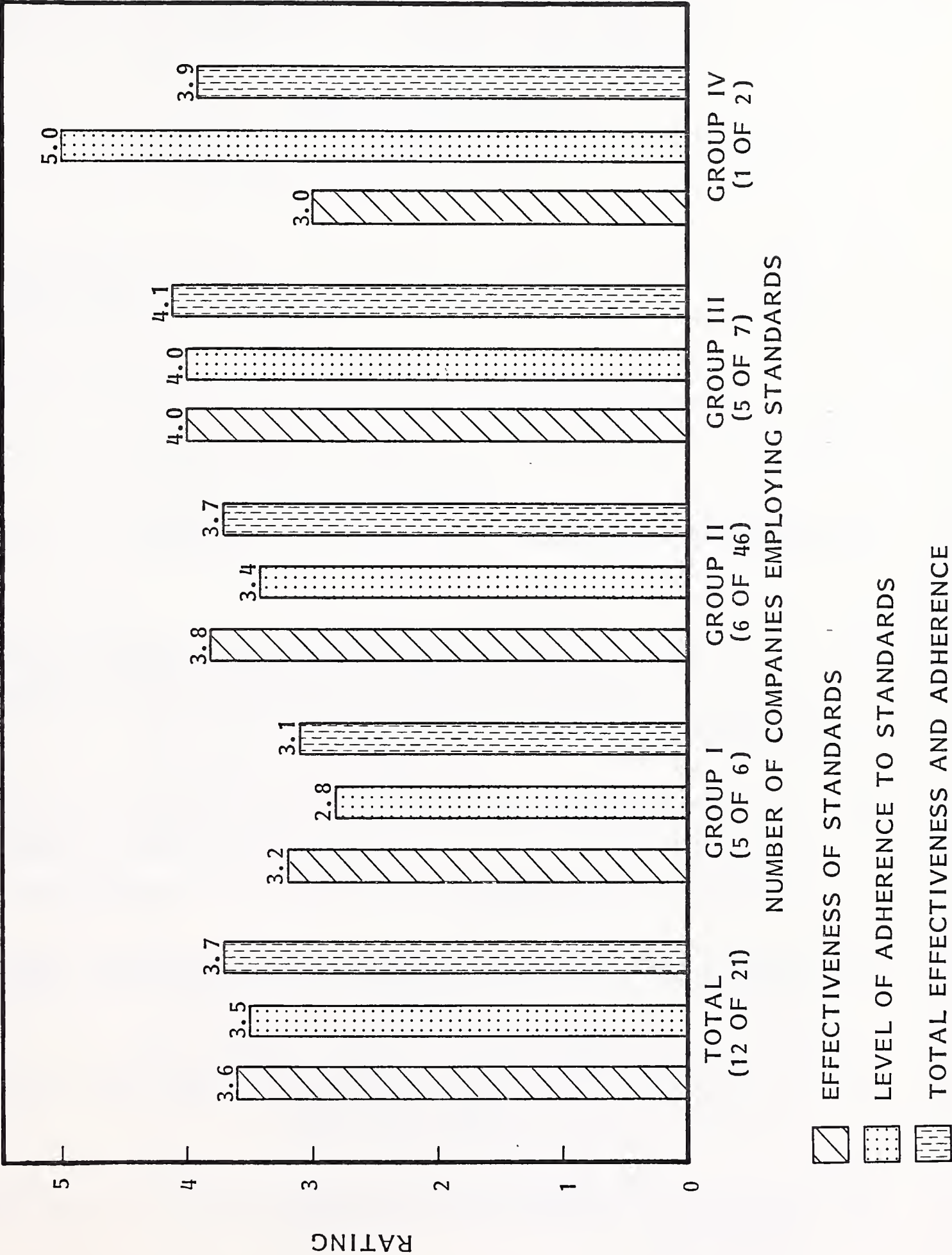
VI STANDARDS AND QUALITY CONTROL IN SOFTWARE DEVELOPMENT

A. PROGRAMMING STANDARDS

- One of the most startling findings in this study is that there may not be a correlation between the use of programming standards and the effective development of software. This finding conflicts with other findings in the study and with generally accepted ideas. It raises some interesting questions that should be explored further.
- The surveyed companies were asked if they used programming standards.
 - Seventeen responded "yes," and 4 responded "no."
 - The group responding "yes" was nearly equal between the companies in Group I and the combined companies in Groups II, III and IV.
- The companies were not given a definition of programming standards.
 - Each company was permitted to interpret the term as it preferred.
 - It is safe to assume that each company used the term in the context of its own software work, and the interpretation was likely to differ from company to company.

- The companies were also asked to rate the effectiveness and level of adherence to those standards on a scale of 1 to 5, where 1 is the least effective or lowest level of adherence and 5 the most effective or highest.
- The ratings shown in Exhibit VI-1 show an inverse correlation to the effectiveness of the software development effort.
- The inverse correlation may not hold any true significance for several reasons:
 - The standard error of deviation is greater than any of the variations shown in the exhibit.
 - If the companies who had no standards were not excluded, but rather assigned ratings of 1 because they did not employ standards, then the inverse correlation would probably disappear.
- Effective standards may not contribute to productivity for several reasons:
 - Standards limit the creative process, and software development is a creative process.
 - Standards are virtually mandatory for large development projects, which are often less efficient than smaller ones. However, there is no indication that standards per se lead to efficiency.
- Any loss in efficiency and productivity in the development process due to standards may be acceptable in light of future savings in software maintenance.
- Most of the Group I companies had separate maintenance groups, which require higher standards.
 - An overwhelming majority of Groups II, III and IV companies did not separate maintenance from development.

USE, EFFECTIVENESS AND ADHERENCE TO STANDARDS



- These companies' ratings of standards may be significantly affected by their need for them, and so may not be very meaningful when comparing between groups.
- The primary thing to be learned from these findings is that the effect of standards on productivity should probably be examined more closely.
- The companies' responses to the question, "If you have standards, how do you measure the level of adherence to those standards?" are shown in Exhibit VI-2.
- From Group I to Group III, the statements become somewhat more quantitative.

B. QUALITY CONTROL

- The surveyed companies were asked how they measured the quality of software development and what type of quality control mechanisms they employed.
- The level of quality control employed by each company was evaluated by counting the number of specific quality control measures mentioned by the respondents to the various questions in Section 4 of the questionnaire.
- The average number of measures mentioned for each group is shown in Exhibit VI-3.
- A high level of quality control is associated with the Group I companies, with much lower levels for the other three groups.
- A high level of quality control is extremely important to effective software development.

EXHIBIT VI-2

RESPONSES TO THE QUESTION,
"IF YOU HAVE STANDARDS, HOW DO YOU
MEASURE THE LEVEL OF ADHERENCE TO
THOSE STANDARDS?"

GROUP I

COMPANY A	"Don't formally measure, informal process, peer pressure."
COMPANY C	"Right now, on feeling - we get bitten and then we'll talk to someone involved."
COMPANY D	"Measure through checkpoints throughout development process"

GROUP II

COMPANY L	"Code reviews and design reviews."
COMPANY M	"I just know over 95% follow standard."
COMPANY N	"By testing - they just are, they're obvious, functional and that's the way it is."
COMPANY O	"By examining program structures that we see."
COMPANY P	"Spot checking."

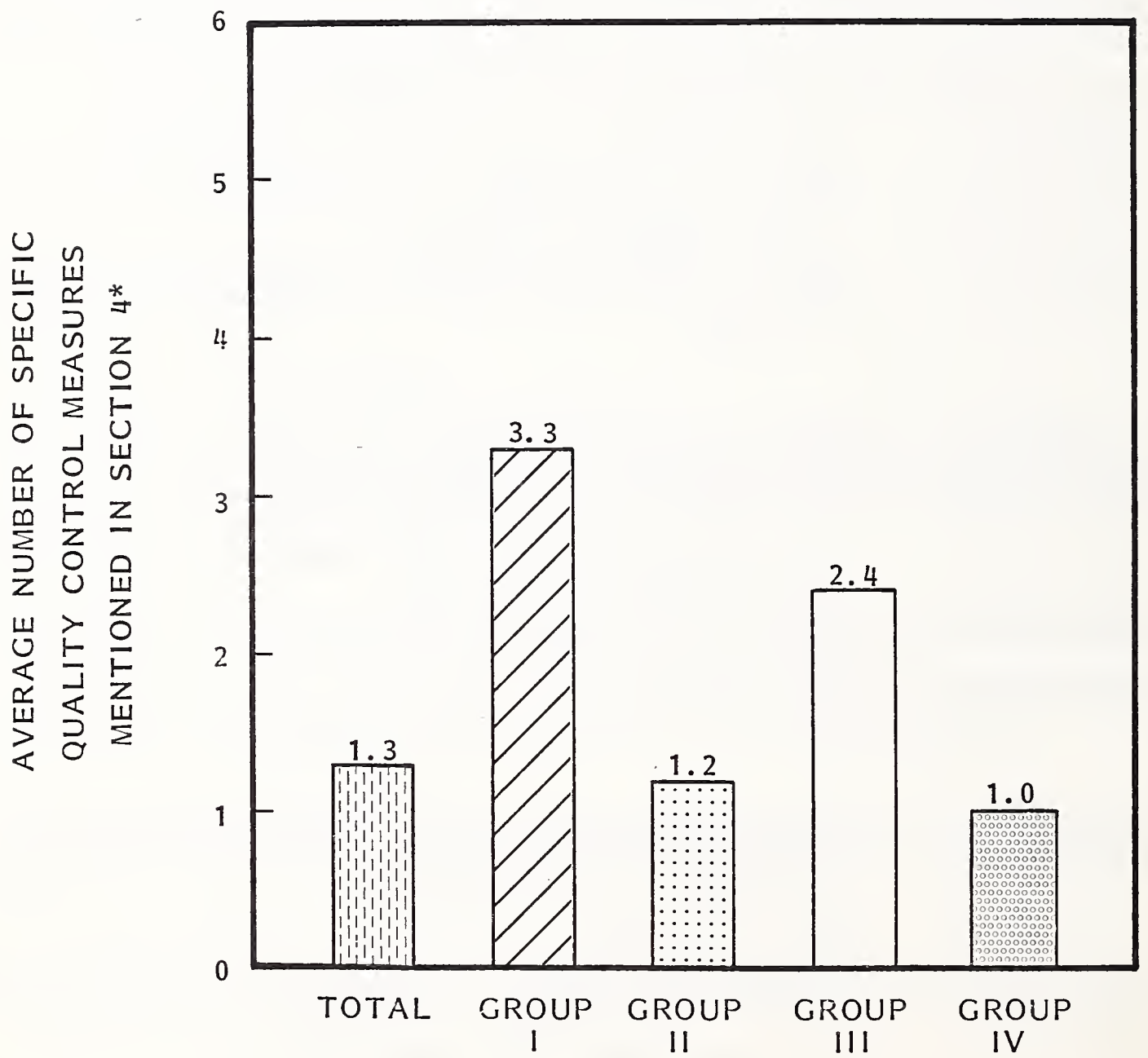
GROUP III

COMPANY H	"No formal measure-problem resolution tells about it. Structured programming enforced by the languages and others by management."
COMPANY I	"New people coming in have not been given training to indoctrinate them in the standards. We rely on people to do it themselves - no policing."
COMPANY K	"Integration good, maintenance good."
COMPANY Q	"Certain people watch this. Software library will review adherence and issue part number or it doesn't become a product."
COMPANY T	"Project managers know coding and how program is coded."

NOTE: GROUP IV HAD NO RESPONSES

EXHIBIT VI-3

LEVEL OF QUALITY CONTROL



* RANGE OF RESPONSES WAS FROM 0 TO 6

- A quality assurance department that reports to high-level management and is separate from the software development department was often found in the more successful companies.
 - These companies also usually combined hardware and software quality control.
- The companies' responses to the question, "How do you measure the quality of your software development?" are shown in Exhibit VI-4.
- The companies in Group I generally employed effective measures that were:
 - Quantitative.
 - Scientific.
 - Statistical.
 - Historical.
 - Predictive.
- Similar measures were also cited when asked "What type of quality control mechanisms do you employ?" as shown in Exhibit VI-5.
- The more successful companies closely related quality control to revenues and profits.
- Establishing goals, in addition to measures, for quality control is also an important factor.
- When asked what procedural improvements they would recommend for increasing software product quality, the companies responded as shown in Exhibit VI-6.

EXHIBIT VI-4
RESPONSES TO THE QUESTION,
"HOW DO YOU MEASURE THE QUALITY OF
YOUR SOFTWARE DEVELOPMENT?"

GROUP I

- | | |
|-----------|--|
| COMPANY A | "Statistics. Bug rate per 1,000 hours of use. Product team puts test plan together to test all features. Bugs per installation per year. Weight bugs for severity of problem." |
| COMPANY B | "No measures." |
| COMPANY C | "A problem because the emphasis is on profits. Trying to develop a test bed and regression test bed for each product. Trying to track failure rate of MTBF." |
| COMPANY D | "Internal procedures. Maintain statistics on problems. Mean time operating statistics. Establish goals and measure performance relative to goals. Monitor problems." |
| COMPANY E | "If the developer says it works. If it satisfies the customer. Try to use predictors rather than measures." |
| COMPANY F | "Customer complaints about bugs." |

GROUP II

- | | |
|-----------|--|
| COMPANY G | "It works - performance." |
| COMPANY L | "Methodology, milestones." "Bug level/severity." |
| COMPANY M | "Are there bugs? Is it competitive? Q/C techniques - Quality assurance specifications by design team. How well it sells." |
| COMPANY N | "Usability is directly related to quality." "Multiple installations by dealers." "If it functions." |
| COMPANY O | "In relationship to how well hardware functions using software; how well it compares to other software on market. How well it accomplishes its objective. Response from distributors." |
| COMPANY P | "We don't have any quantitative measures. With great difficulty." |

EXHIBIT VI-4 (CONT.)
RESPONSES TO THE QUESTION,
"HOW DO YOU MEASURE THE QUALITY OF
YOUR SOFTWARE DEVELOPMENT?"

GROUP III

COMPANY H	"Market acceptance of the product." "Budgets are not a constraint because it's such a small part of total cost." "Schedules are important."
COMPANY I	"We have a test and maintenance group that do alpha and beta test. Customer acceptance."
COMPANY K	"If it's on time and on budget." "If it functions." "Customer acceptance."
COMPANY Q	"Does it function in regards to the documentation and internal specifications?" "Quality of external (user) documentation." "Efficiency of performance." "Modularity, reliability."
COMPANY R	"Release for alpha test - get results." "Try to fix bugs on next release, no more than six weeks." "Sample programs, regression tests." "Customer exercises product before it's handed over to end user."
COMPANY S	"If it works and supports people in doing what they have to do."
COMPANY T	"Good reports from the field."

GROUP IV

COMPANY J	"Whether it performs the function the user needs. Is it timely? Can it be delivered by someone other than the inventor?"
COMPANY U	"Customer satisfaction."

EXHIBIT VI-5

RESPONSES TO THE QUESTION,
"WHAT TYPE OF QUALITY CONTROL
MECHANISM DO YOU EMPLOY?"

GROUP I

COMPANY B	"Quality assurance test it."
	"Test bed,
COMPANY C	Test programmer support.
	Require detailed design specs.
	Final test and approval (handoff)."
	"Audit internal mechanism.
COMPANY E	Internal procedures.
	Walk-throughs.
	Read code.
	Impose discipline in development process."
COMPANY F	"New product reviews based on specification."

GROUP II

COMPANY L	"Methodology, acceptance testing, maintenance reports."
COMPANY N	"Primarily adequate testing. In-house testing of everything before release. At least two dealers field test and feedback results to us."
COMPANY P	"They are all new. They are starting to work. We use a specific quality assurance person."

GROUP III

COMPANY K	"Maintainable and reliable."
COMPANY Q	"Prototype functional test by quality assurance. Quality assurance is involved from bread board stage to full production. Schedule and plans."
COMPANY R	"In-house we do alpha test. We have a formal quality control group. We perform regression test."
COMPANY S	"I look at stuff and say how it should be done. Do things myself. Developing a style book."
COMPANY T	"We work closely with marketing. Going through a design phase. There is tight control. Try to meet marketing specification. Quality assurance systems and sign-off procedures."

EXHIBIT VI-5 (CONT.)

RESPONSES TO THE QUESTION,
"WHAT TYPE OF QUALITY CONTROL
MECHANISM DO YOU EMPLOY?"

GROUP IV

COMPANY J	"Family of acceptance and regression tests. Each generation is run against these tests."
COMPANY U	"None."

EXHIBIT VI-6

RESPONSES TO THE QUESTION,
 "WHAT PROCEDURAL IMPROVEMENTS WOULD
 YOU RECOMMEND FOR INCREASING SOFTWARE
 PRODUCT QUALITY?"

RECOMMENDATIONS	GROUPS				
	TOTAL	GROUP I	GROUP II	GROUP III	GROUP IV
IMPROVE INTERFACE BETWEEN SPECIFICATION AND DESIGN	1	1	-	-	-
POLICE SPECIFICATIONS	3	1	2	-	-
WRITTEN SPECIFICATIONS OR GUIDELINES	2	1	-	-	1
BETTER DEFINITION OF INTEGRATION AND HAND-OFF PROCEDURES	2	1	1	-	-
DEVELOP MORE DEVELOPMENT TOOLS	3	3	-	-	-
MORE REGRESSION TESTING	1	1	-	-	-
DEVELOP SELF-TESTING SOFTWARE	1	1	-	-	-
USE FEATURE FUNCTION TEST	1	1	-	-	-
IMPLEMENT STANDARDS	2	1	1	-	-
IMPROVE DOCUMENTATION	1	1	-	-	-
MOTIVATIONAL TRAINING FOR MANAGERS	1	1	-	-	-
IMPROVE ENVIRONMENT	1	1	-	-	-
CHECKS AND BALANCES FOR PRODUCT RELEASE	1	-	1	-	-
IMPLEMENT PROFITSHARING					
USE A HIGHER-LEVEL LANGUAGE	1	-	-	-	1
PAY OVERTIME	1	-	1	-	-
MULTIPLE REVIEW AT ALL STAGES	1	-	1	-	-
IMPROVE PRODUCT DEFINITION IN MARKETING GROUP	2	-	-	2	-
STRONGER QUALITY ASSURANCE	2	-	-	2	-
TOTAL	27	14	7	4	2

- The most frequently mentioned recommendation involved improving specifications or the specification process.
- Improvements in quality assurance were also frequently mentioned.
- Several companies suggested that implementing standards would improve product quality.

C. MEASURES OF SOFTWARE DEVELOPMENT SUCCESS

- The companies indicated that they measured software development success and failure in more than 20 different ways, as shown in Exhibit VI-7.
- The most common measures, in order of frequency, were:
 - Product developed on schedule.
 - Customer acceptance of product.
 - Product works.
 - Product was developed within its budget.
 - Product had high sales volume.
 - Product meets specifications.
- Many of the measures mentioned were not measures of a successful product but of the product development process.
- The more successful companies in Group I tended to measure the product more than the process.

EXHIBIT VI-7

RESPONSES TO THE QUESTION,
 "HOW DO YOU MEASURE SOFTWARE DEVELOPMENT
 SUCCESS AND FAILURE?"

MEASURE OF SUCCESS AND FAILURE	NUMBER OF RESPONDENTS REPORTING THIS MEASUREMENT				
	TOTAL	GROUP I	GROUP II	GROUP III	GROUP IV
DEVELOPED ON SCHEDULE	11	4	1	5	1
CUSTOMER ACCEPTANCE	9	2	1	5	1
PRODUCT FUNCTIONS	7	2	1	3	1
DEVELOPED ON BUDGET	5	2	1	2	-
SALES VOLUME	5	2	1	2	-
PRODUCT MEETS SPECIFICATIONS	4	1	2	1	-
LEVEL OF MAINTENANCE	3	1	1	1	-
PRODUCT PERFORMANCE	2	-	1	1	-
IN-HOUSE TESTING	2	-	2	-	-
SALES FORCE REACTION	2	-	1	1	-
ACCURATE FORECAST OF EFFORT	1	1	-	-	-
QUALITY OF CODE	1	1	-	-	-
DOCUMENTATION AVAILABLE	1	1	-	-	-
GOOD DOCUMENTATION	1	1	-	-	-
PRODUCT OBSOLETE BEFORE RELEASE	1	1	-	-	-
FORECASTED PROBLEM RATE VERSUS ACTUAL PROBLEM RATE	1	1	-	-	-
RETURN ON INVESTMENT	1	1	-	-	-
NUMBER OF INSTALLATIONS	1	-	1	-	-
RELIABILITY	1	-	-	1	-
NO SEVERE CUSTOMER PROBLEMS	1	-	-	1	-
NUMBER OF COMPLAINTS	1	-	-	1	-
PROGRAMMER SATISFIED	1	-	-	1	-
TOTAL	62	21	13	25	3

- Group I companies were concerned about the most important measures of product success, which were:
 - Return on investment in the software product.
 - Customer satisfaction as indicated by:
 - . Increased sales.
 - . Number of installations.
 - . Number of customer complaints.
- The companies were asked, "What software development successes and failures have you experienced within the last five years?"
 - Their responses were analyzed to determine the reasons behind the successes and failures, as shown in Exhibit VI-8.
- In sum, the important and most frequently cited reasons for success were:
 - Having well-conceived and complete specifications:
 - . Which came through good planning and communications with end users and all persons in the company who would come in contact with the product.
 - Producing a thorough and well-designed product.
 - Good controls by management throughout the development process from initial needs assessment of prospective customers to final delivery and maintenance of the product.

EXHIBIT VI-8

REASONS FOR SOFTWARE DEVELOPMENT SUCCESSES
AND FAILURES IN PAST FIVE YEARS

REASONS	GROUPS				
	TOTAL	I	II	III	IV
WELL CONCEIVED AND COMPLETE SPECIFICATIONS	6	3	2	1	-
THROUGH, WELL-DESIGNED (NOT OVER-DESIGNED) PRODUCT	4	3	1	-	-
OUT OF TOUCH WITH MARKETPLACE	3	1	-	2	-
CHANGES IN SPECIFICATIONS	1	1	-	-	-
IMPOSSIBLE SPECIFICATIONS	1	-	-	-	1
COMPETENT DEVELOPMENT PEOPLE	1	1	-	-	-
COMPETENT MANAGEMENT PEOPLE	1	-	1	-	-
TOO MANY PEOPLE	1	1	-	-	-
SELECT RIGHT PEOPLE FOR JOB	1	-	1	-	-
DEMAND HIGH PERFORMANCE	1	-	1	-	-
GOOD CONTROL OF DEVELOPMENT	2	-	1	1	-
USE OF DEVELOPMENT TOOLS	1	-	-	1	-
TOO MUCH FUNCTIONALITY IN THE HARDWARE	1	-	1	-	-
FORCED TO USE POOR (PARENT COMPANY'S) HARDWARE	1	-	-	1	-
TOTAL	25	10	8	6	1

- Good software people and good management people were cited as making the difference in many cases.
 - . It takes good people to attract and hold good people.
 - . It takes a few good people to produce software, not a large team.
- Contact with end users is probably the single most important reason for software development success.
 - . It was evident in all of the most effective software development groups that personal contact with the customer was a vital ingredient in their success.
 - . It helped the software people better understand the customer's problem.
 - . It was a constant reminder of why the software people were doing what they were doing.
- The top companies depended heavily on ego motivators for their software development people. They did this through:
 - . Providing means for the software people to experience the user's appreciation of their work.
 - . Establishing an environment which fostered an identification with the product by the software development people.
 - . Corporate recognition of the contributions made to company revenue and profit goals by the software products and their developers.

- The most frequent reasons for software development failure were:
 - The software developers were out of touch with the marketplace.
 - They didn't understand the needs of the customer.
 - They didn't know the nature of the competition.
 - Problems in specification process:
 - Specifications were either poor, impossible or constantly changing.
 - Specifications were subject to inadequate reviews.
 - The right people did not sign off the specification.
 - Assigning too many people to a project.
 - Over-designing a product.
 - Creating a product too large for the hardware.
 - Providing more product than the customer was willing to pay for.
 - Lack of controls by management throughout the development process.
 - Lack of measures of quality, reliability, maintainability, productivity, profitability, schedules, budgets and customer acceptance.
 - Lack of sign-offs and reviews at all stages of development, which involved all personnel who had anything to do with the product.

VII THE FUTURE OF SOFTWARE DEVELOPMENT

VII THE FUTURE OF SOFTWARE DEVELOPMENT

- The companies were asked which software development methodologies they envisioned being available by 1985 and which ones they expected to implement. Most companies gave the same responses to both questions, indicating that they planned to implement everything they expected to be available.
- Since many of the companies interviewed were leading-edge developers of computer systems, they were often well on the way to implementing many of the methodologies for their advanced systems.
- The most frequently mentioned methodologies are shown in Exhibit VII-1 in order of frequency.
- The most likely methodology to be available will be computer systems and software, which in effect write their own software under the general direction of the end user.
- Much of the software will be incorporated into the hardware as firmware.
- The computers and people will write programs in, or similar to, PASCAL and "C."

EXHIBIT VII-1
SOFTWARE DEVELOPMENT METHODOLOGY RESPONDENTS
ENVISION BEING AVAILABLE BY 1985

TYPE	DESCRIPTION	TOTAL	GROUPS			
			I	II	III	IV
PRO- GRAMMING TOOLS	TOOLS FOR LINKING DEVELOPMENT AND MAINTENANCE	1	1	-	-	-
	MORE PROGRAMMING TOOLS	2	1	-	1	-
	SOFTWARE LIBRARIES OF RE-USABLE CODE	5	1	1	3	-
	MORE AUTOMATED IMPLEMENTATION	1	1	-	-	-
	TOOLS FOR CROSS-CHECKING	1	1	-	-	-
	TOOLS FOR DEVELOPING MODULES FROM DESIGN SPECIFICATIONS	1	1	-	-	-
	SUBTOTAL	11	6	1	4	0
PROGRAM GEN- ERATORS	3 TO 4 COMPILERS USING CODE GENERATORS	1	1	-	-	-
	PROGRAM GENERATORS BY EXAMPLE SPECIFICATION-ORIENTED INTER-ACTIVE TOOLS AND SELF-GEN PGS	1	1	-	-	-
	COMPILER GENERATORS	6	1	1	4	-
	MORE PROGRAM GENERATORS	1	-	1	-	-
		2	-	2	-	-
	SUBTOTAL	11	3	4	4	0
STANDARDS	MORE STRUCTURED PROGRAMMING AND DEVELOPMENT	6	1	2	2	1
	STANDARDS AT LOWER LEVELS OF SYSTEMS DEVELOPMENT	2	1	-	1	-
	BUILT-IN STANDARDS IN LANGUAGE	1	1	-	-	-
	MULTI-USER DEVELOPMENT MACHINES TO ENFORCE STANDARDS	1	-	1	-	-
	SUBTOTAL	10	3	3	3	1
LANGUAGES	HIGH-LEVEL DEVELOPMENT LANGUAGE	5	1	-	3	1
	HIGHER-LEVEL DEBUGGER	1	-	-	-	1
	SUBTOTAL	6	1	0	3	2
DOCU- MENTATION	COMPUTERIZED CHECKING FOR DOCUMENTATION	1	1	-	-	-
	MIXED GRAPHICS AND WORD PROCESSING FOR TECHNICAL DOCUMENTATION	1	-	-	1	-
	SUBTOTAL	2	1	0	1	0
OTHER	STUDYING ORGANIZATIONAL STRUCTURES' IMPACT ON PRODUCTIVITY	1	1	-	-	-
	MORE SOFTWARE IN HARDWARE	2	-	1	1	-
	EXTENSIONS OF FILE MANAGEMENT SYSTEMS AND TRANSACTION MANAGEMENT SYSTEMS	1	-	1	-	-
	SUBTOTAL	4	1	2	1	0
	GRAND TOTAL	44	15	10	16	3

- The languages employed will have built-in standards and will operate on virtually any hardware developed in the 1980s.
- Most programs will document themselves.
- Systems will have comprehensive built-in diagnostics and repair mechanisms for both hardware and software.
- Many problems being addressed will go beyond human comprehension because of the millions of variables and parameters being addressed.
- The number of software people available will not be as critical a problem as the quality of those people.
- By the 1990s, the cost of software will come down relative to hardware as end users are able to verbally instruct their machines, which will eliminate the programmer from the development process.
 - The machines will actually be able to suggest a number of alternative solutions to any given problem, list the advantages and disadvantages of each option, and recommend the best solution.
- A likely scenario would be:
 - Early 1980s.
 - Programming tools.
 - Software libraries on chips.
 - High-level structured programs.

- . Built-in standards.
- Mid 1980s.
 - . Program generators.
 - . Compiler generators.
 - . System generators (software and hardware).
- The 1990s.
 - . Automatic software and hardware designing systems.
 - . Systems completely compatible and "friendly" with end users.
 - . Literally conversational systems with significant creative capabilities.

APPENDIX A: INTERVIEW PROFILES

APPENDIX A: INTERVIEW PROFILES

- Anderson Jacobson, Inc.
- Apple Computer Inc.
- BRD (Bainbridge Research & Development), Inc.
- Centurion Corporation
- Century Computer Corporation
- Display Data Corporation
- Digital Telephone Systems, Inc. (subsidiary of Harris Corp.)
- Fairchild Camera and Instrument Corporation
- Four-Phase Systems, Inc.
- General Automation
- Harris Corporation - Computer Systems Division
- Hewlett Packard General Systems Division
- Mitsubishi Electronics (Melcom Business Systems)

- NCR
- New England Digital Corporation
- Olivetti Corporation of America
- Ontel
- Raytheon Data Systems Company
- Tymshare, Inc.
- Western Electric Corporation
- Zilog (subsidiary of Exxon Corporation)

APPENDIX B: DEFINITIONS

APPENDIX B: DEFINITIONS

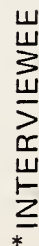
- **BUSINESS UNIT SOFTWARE ORGANIZATION** contains all of the functions, or representatives of the functions, necessary to produce and market a product (including manufacturing, marketing, planning and quality assurance) assigned to a development group.
 - The business unit is supposed to have all of the capabilities of a company.
 - This type was categorized as a project organization for this study.
- **FUNCTIONAL SOFTWARE ORGANIZATION** is organized so that individuals are assigned to specific functions that cross product lines.
- **PROJECT SOFTWARE ORGANIZATION** is organized so that individuals are assigned to specific products and may perform a variety of functions relating to that product.
- **MATRIX SOFTWARE ORGANIZATION** is organized so that individuals share responsibility and/or authority for developing the software with other individuals outside of the software development organization.

APPENDIX C: DATA BASE OF ORGANIZATIONAL
CHARTS

APPENDIX C: DATA BASE OF ORGANIZATIONAL CHARTS

- The organizational charts are presented in alphabetical order.
 - Charts A through E are the five public Group I companies.
 - Chart F is the private Group I company.
 - Charts G through K are the five public companies from Groups II, III and IV.
 - G and J are in Group II.
 - H, I and K are in Group III.
 - Charts L through P are in Group II.
 - Charts Q through T are in Group III.
 - Chart U is in Group IV.

INPL



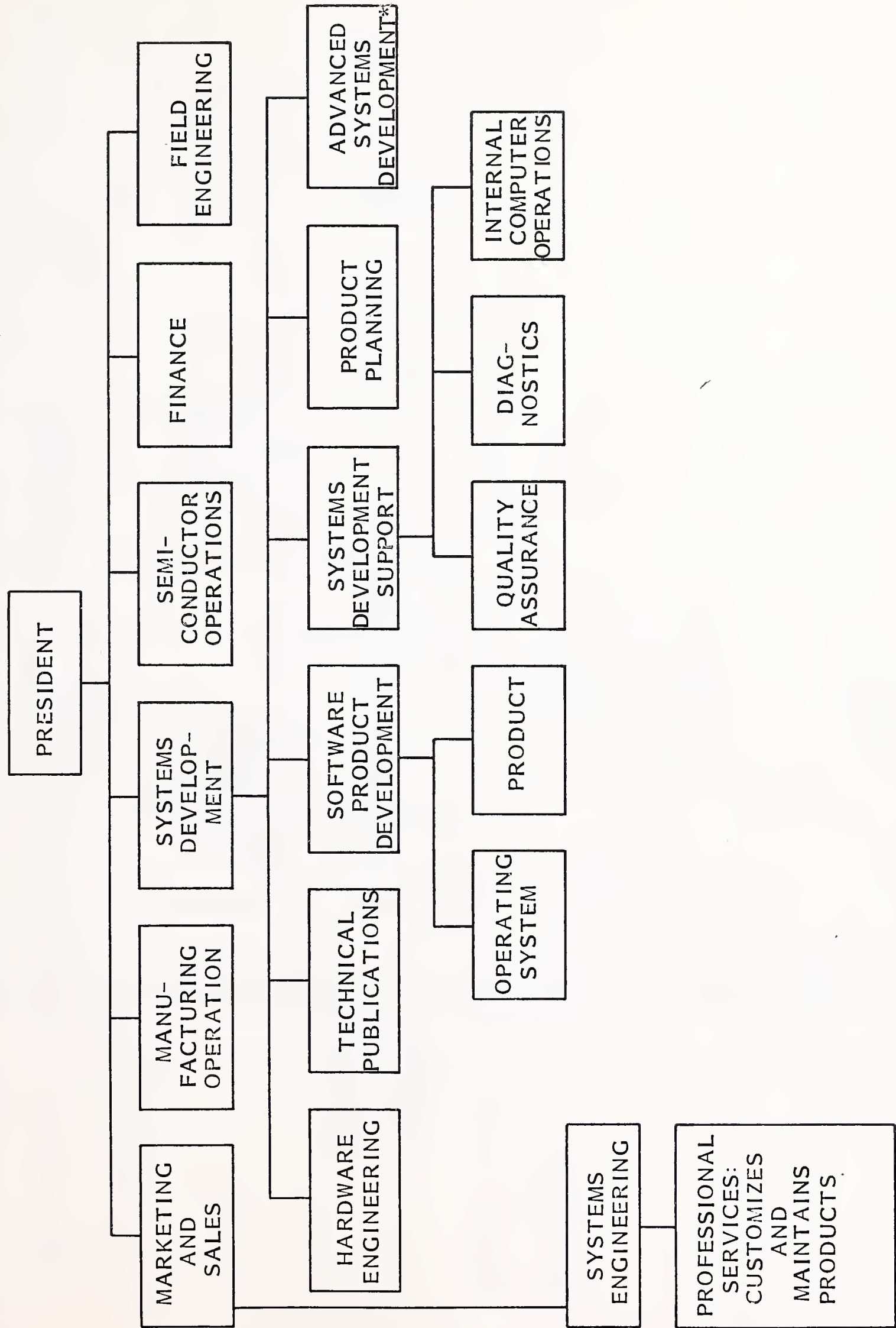


EXHIBIT C-3
COMPANY C

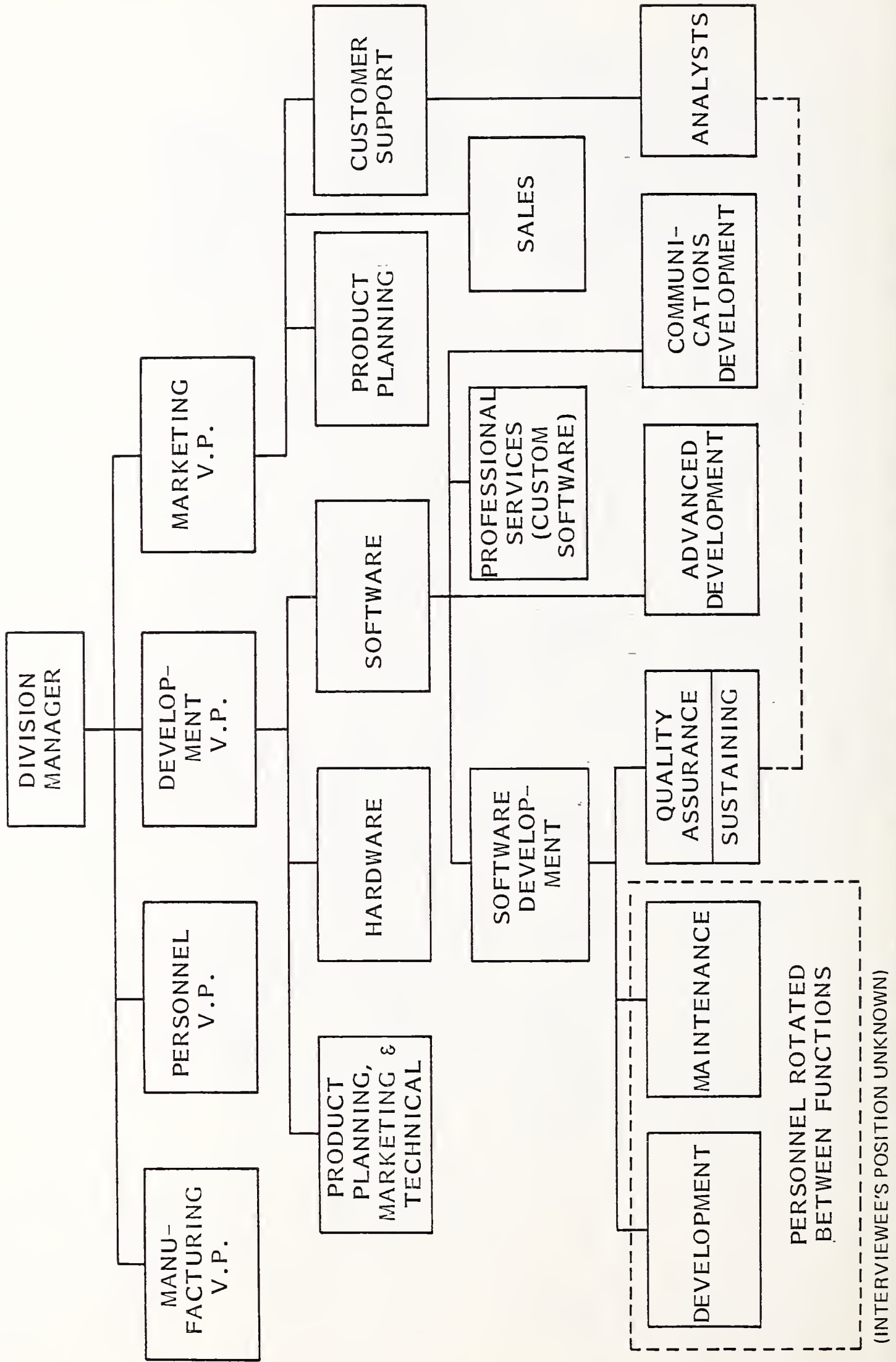
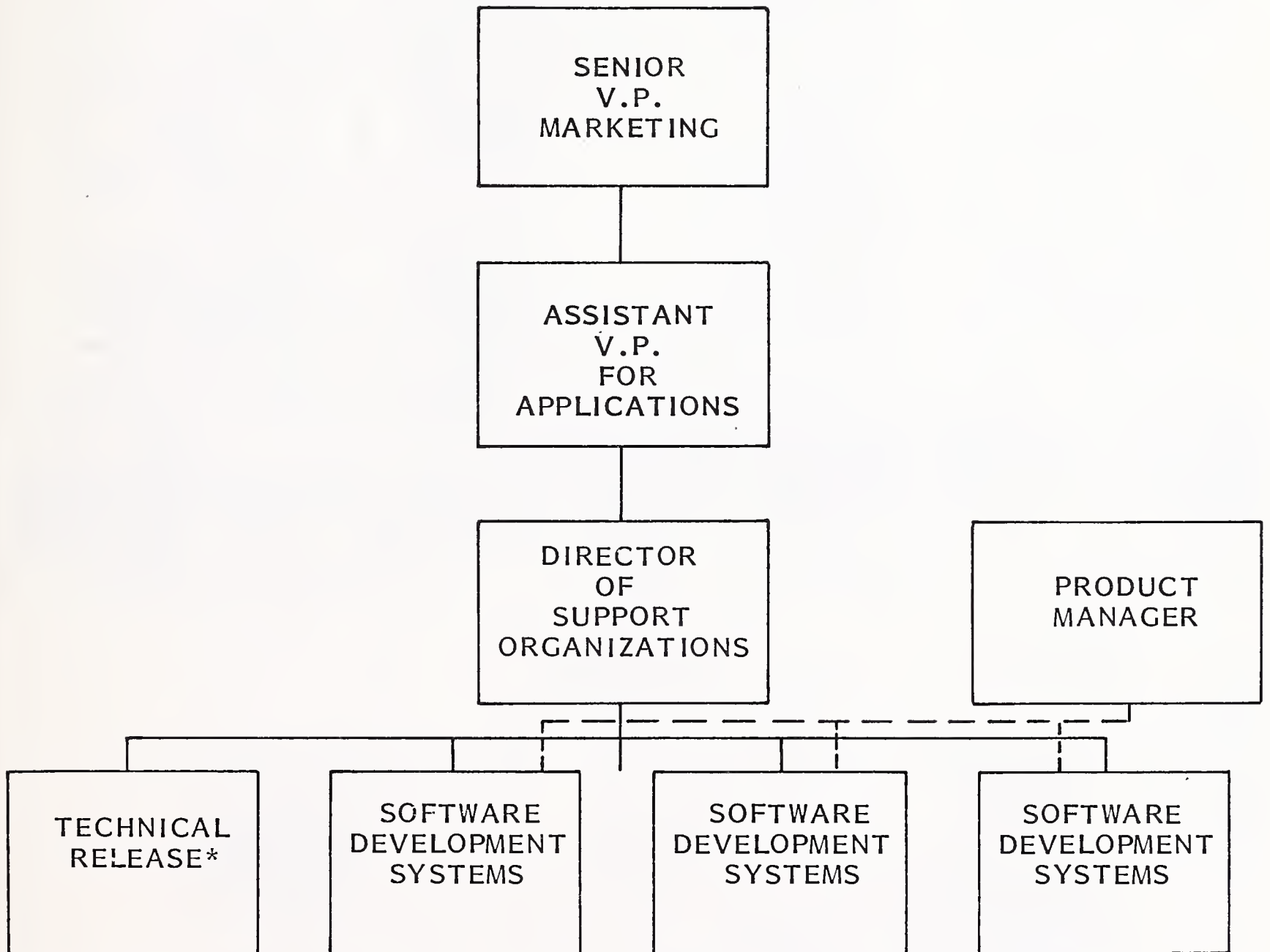


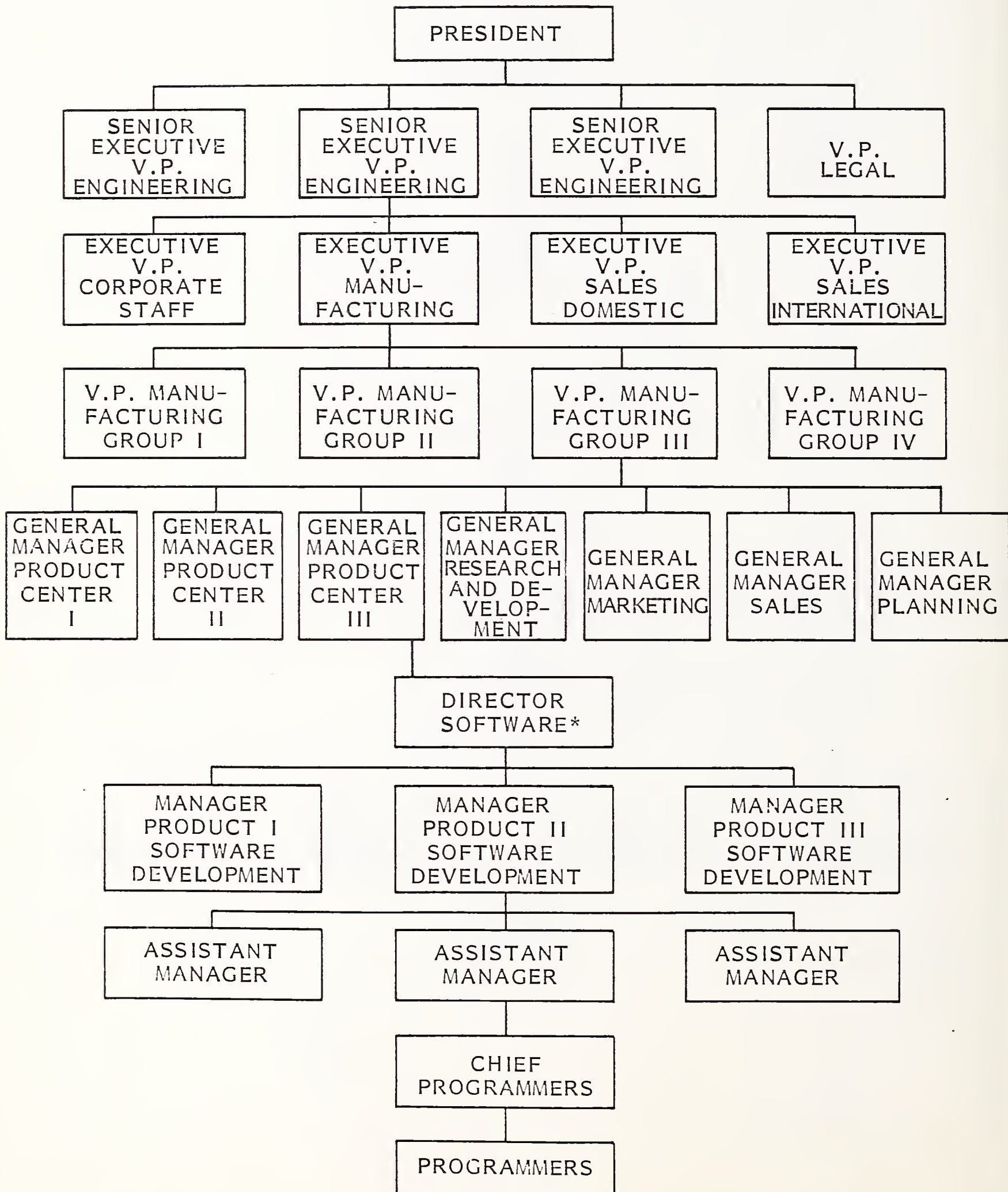
EXHIBIT C-4

COMPANY D

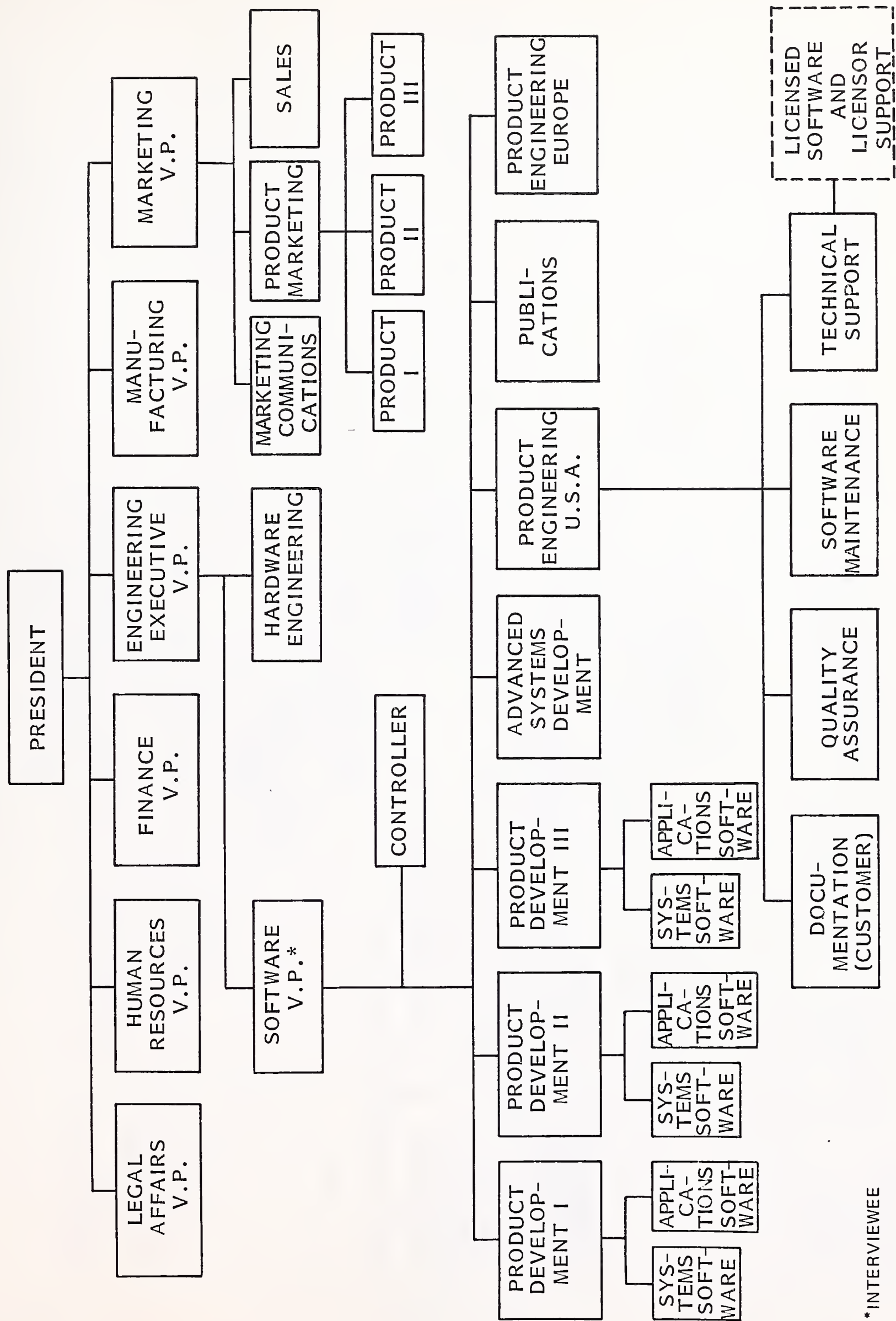


*INTERVIEWEE

EXHIBIT C-5
COMPANY E

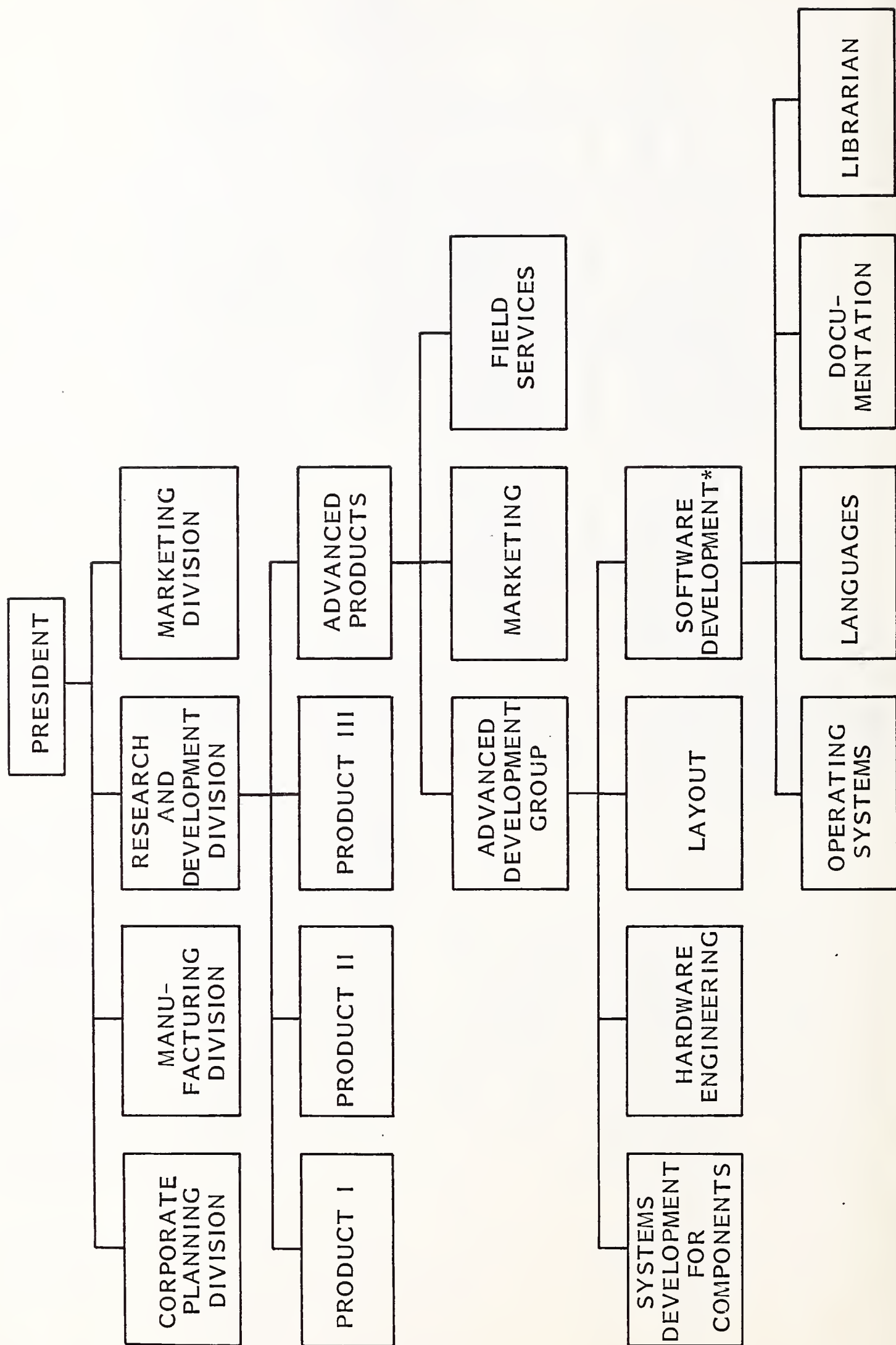


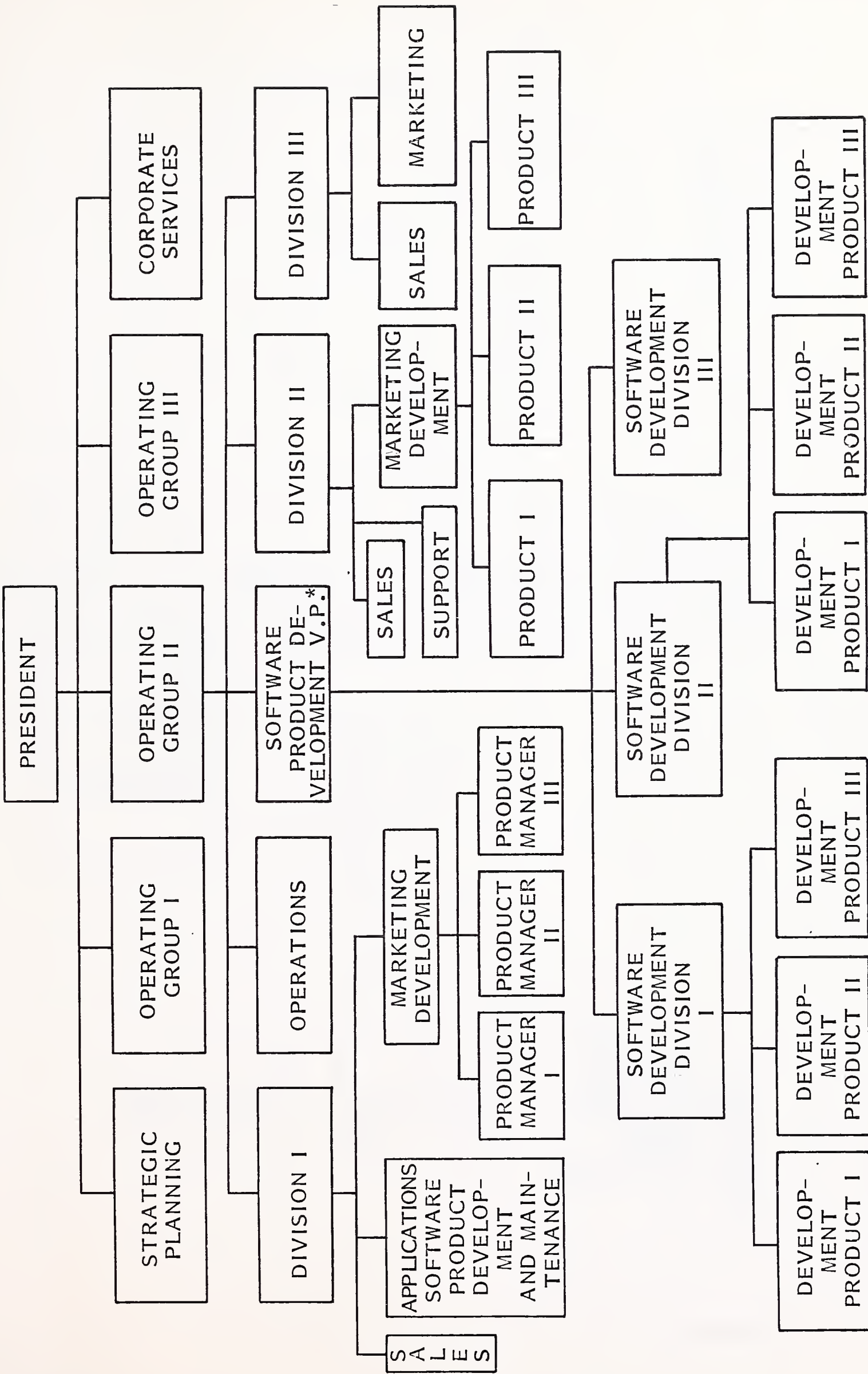
*INTERVIEWEE



* INTERVIEWEE

EXHIBIT C-7
COMPANY G

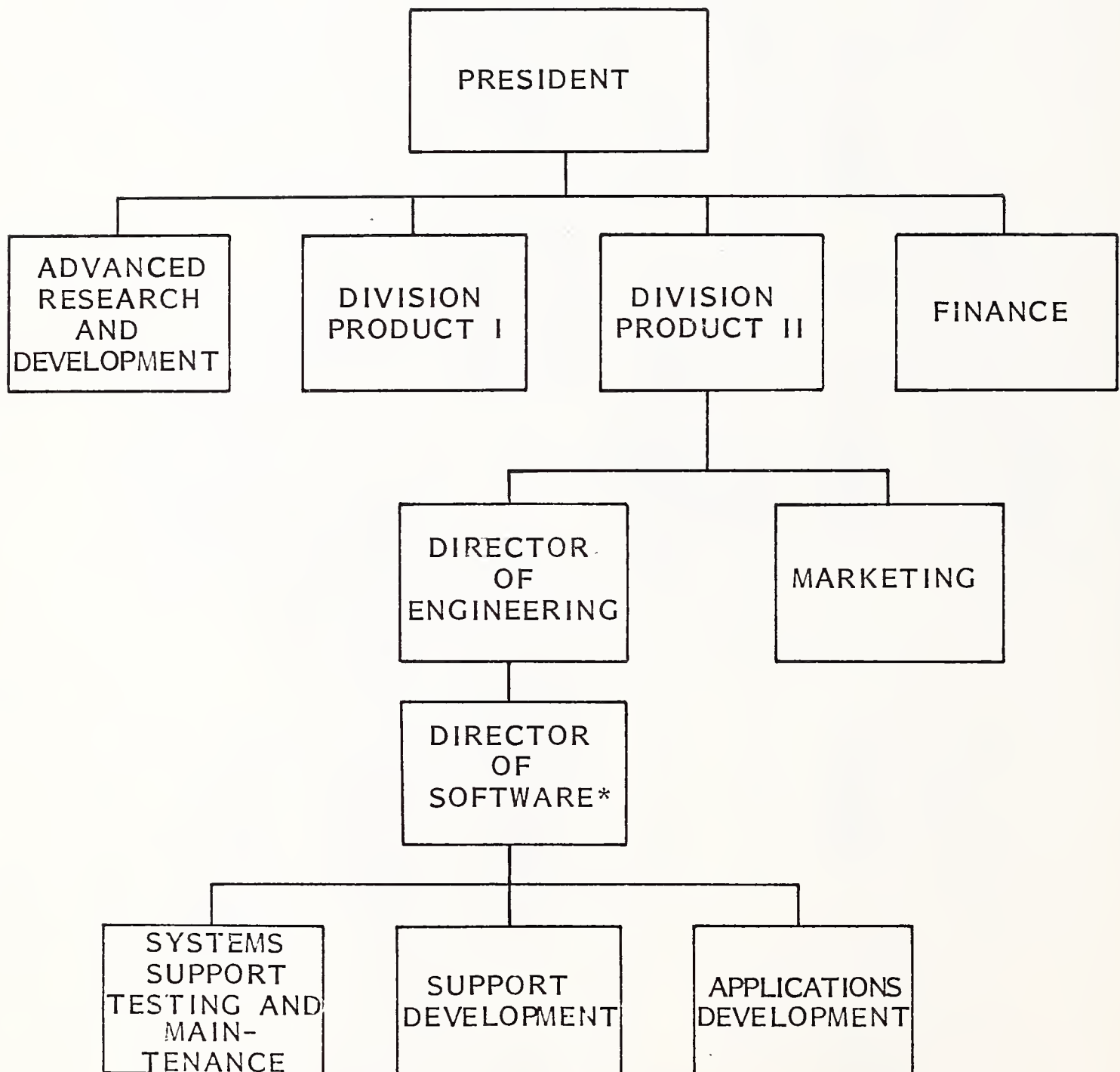




*INTERVIEWEE

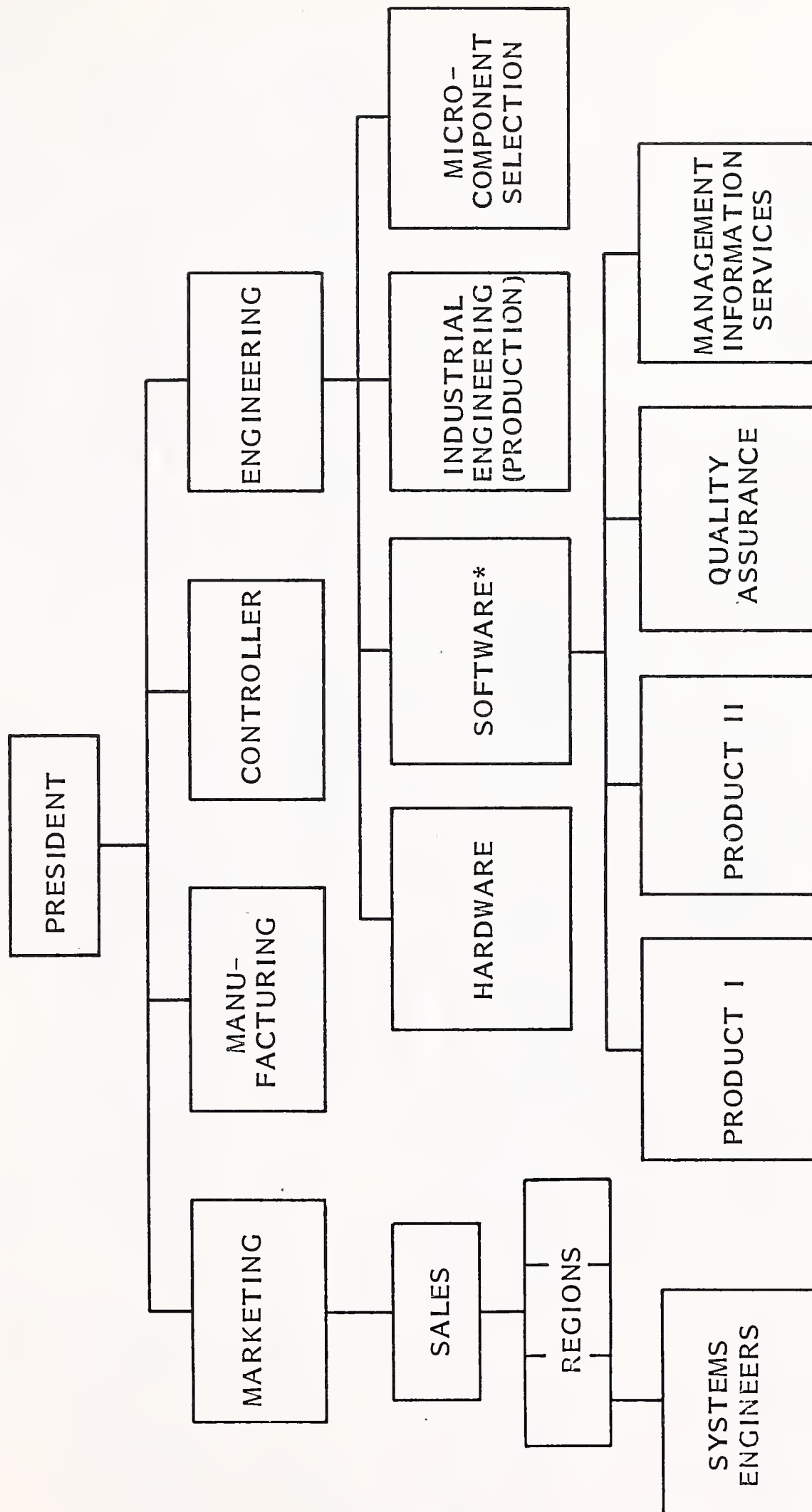
EXHIBIT C-9

COMPANY I

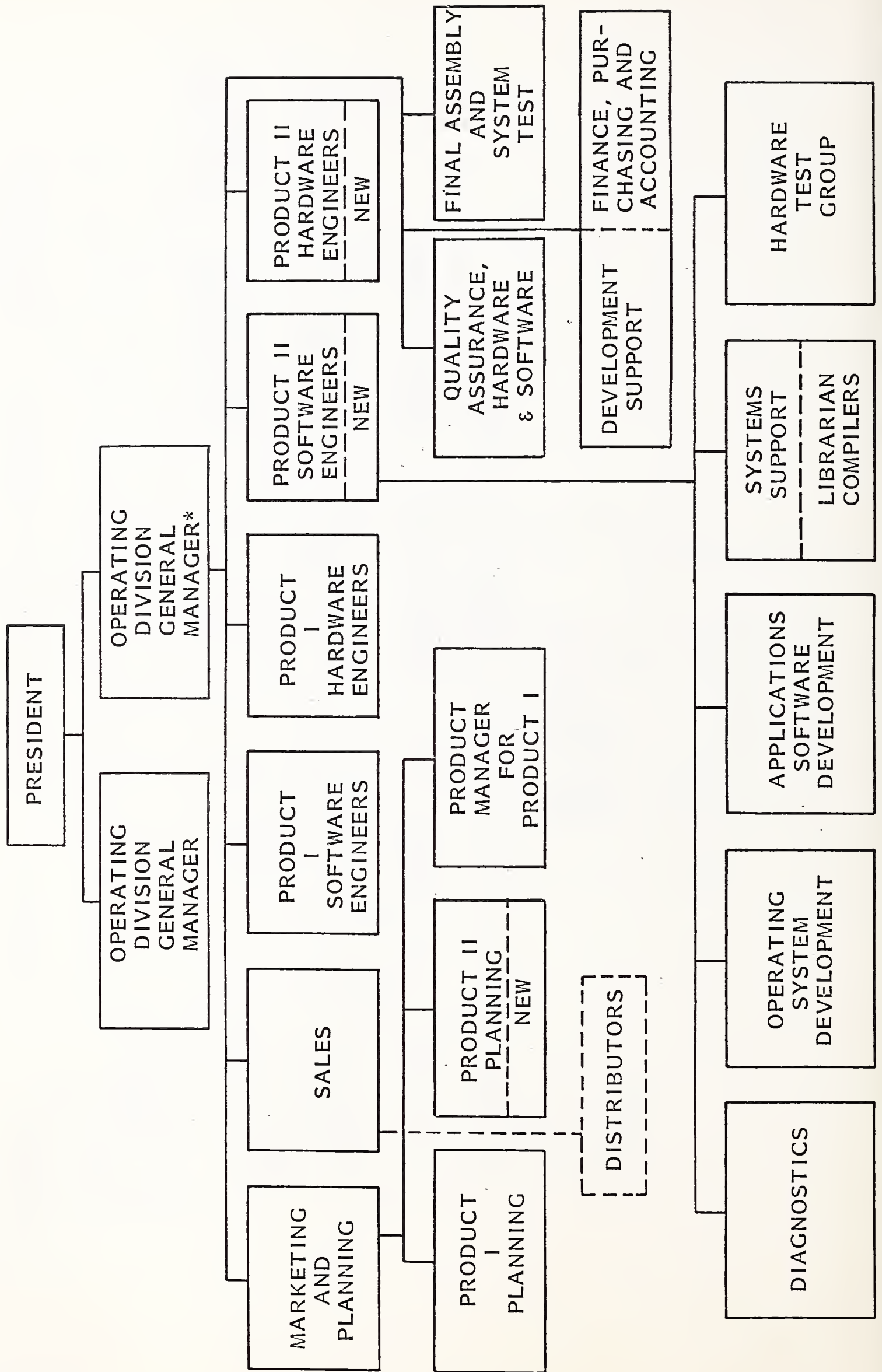


*INTERVIEWEE

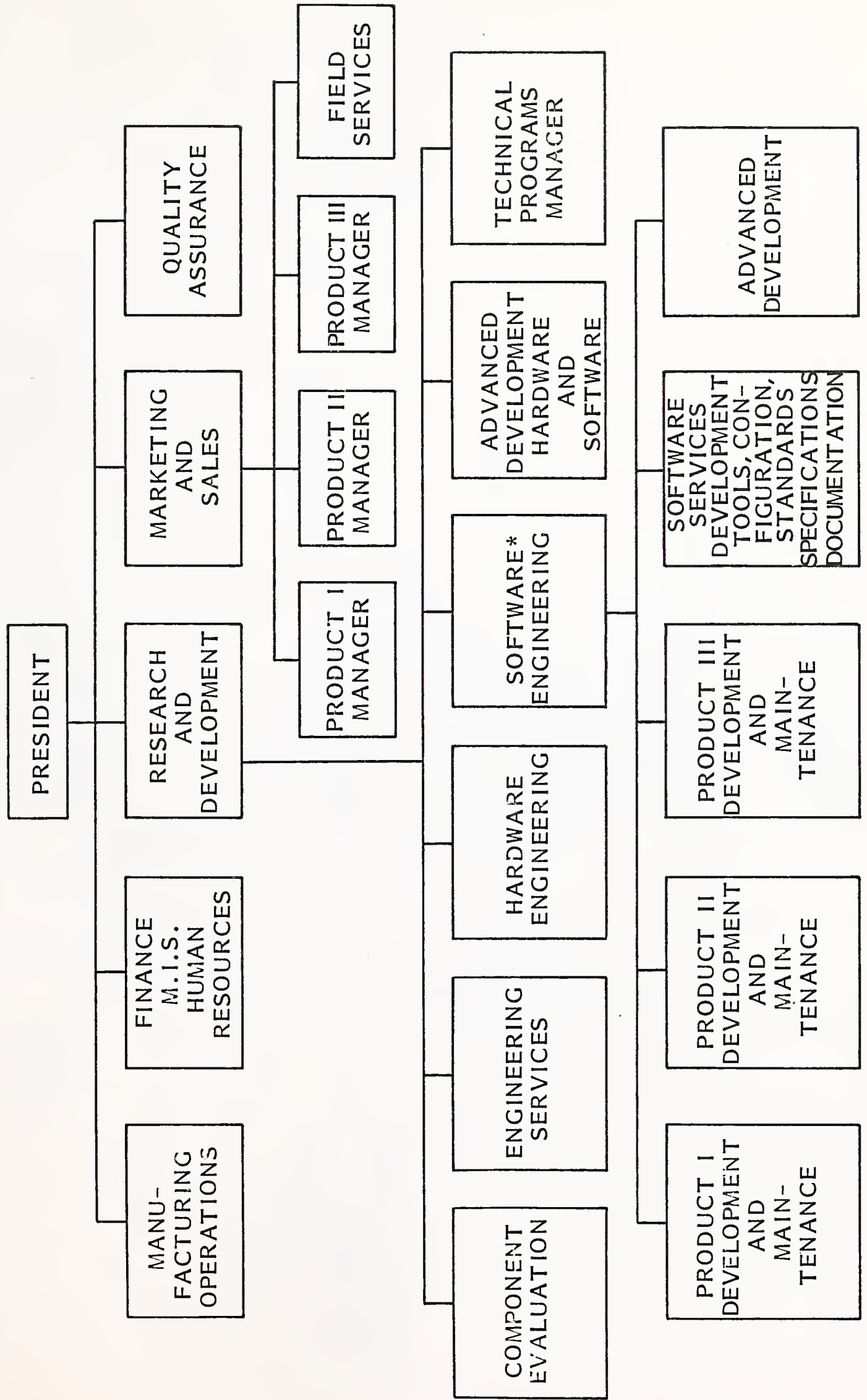
COMPANY J



* INTERVIEWEE



COMPANY L



*INTERVIEWEE

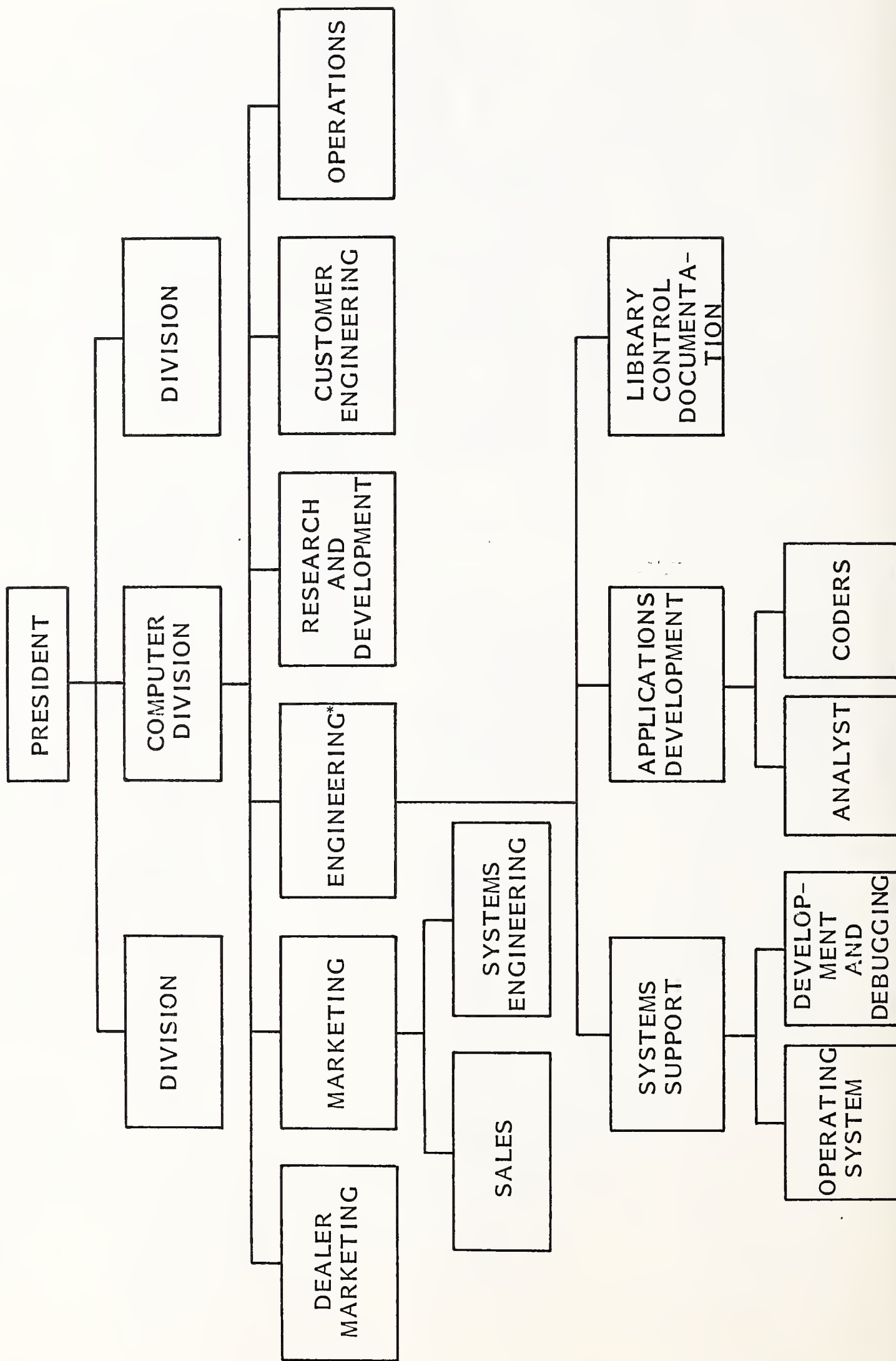
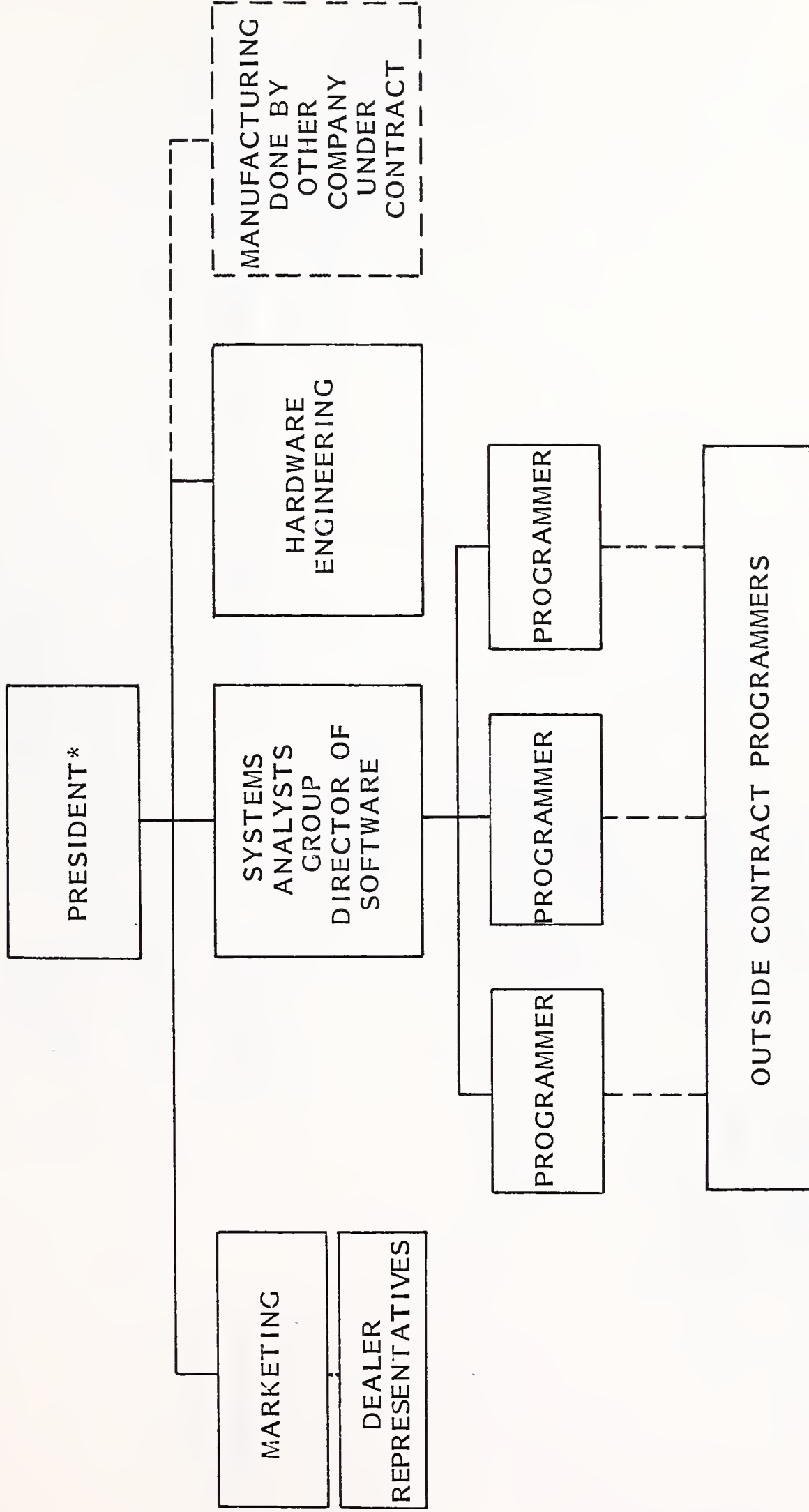


EXHIBIT C-14

COMPANY N



* INTERVIEWEE

EXHIBIT C-15
COMPANY O

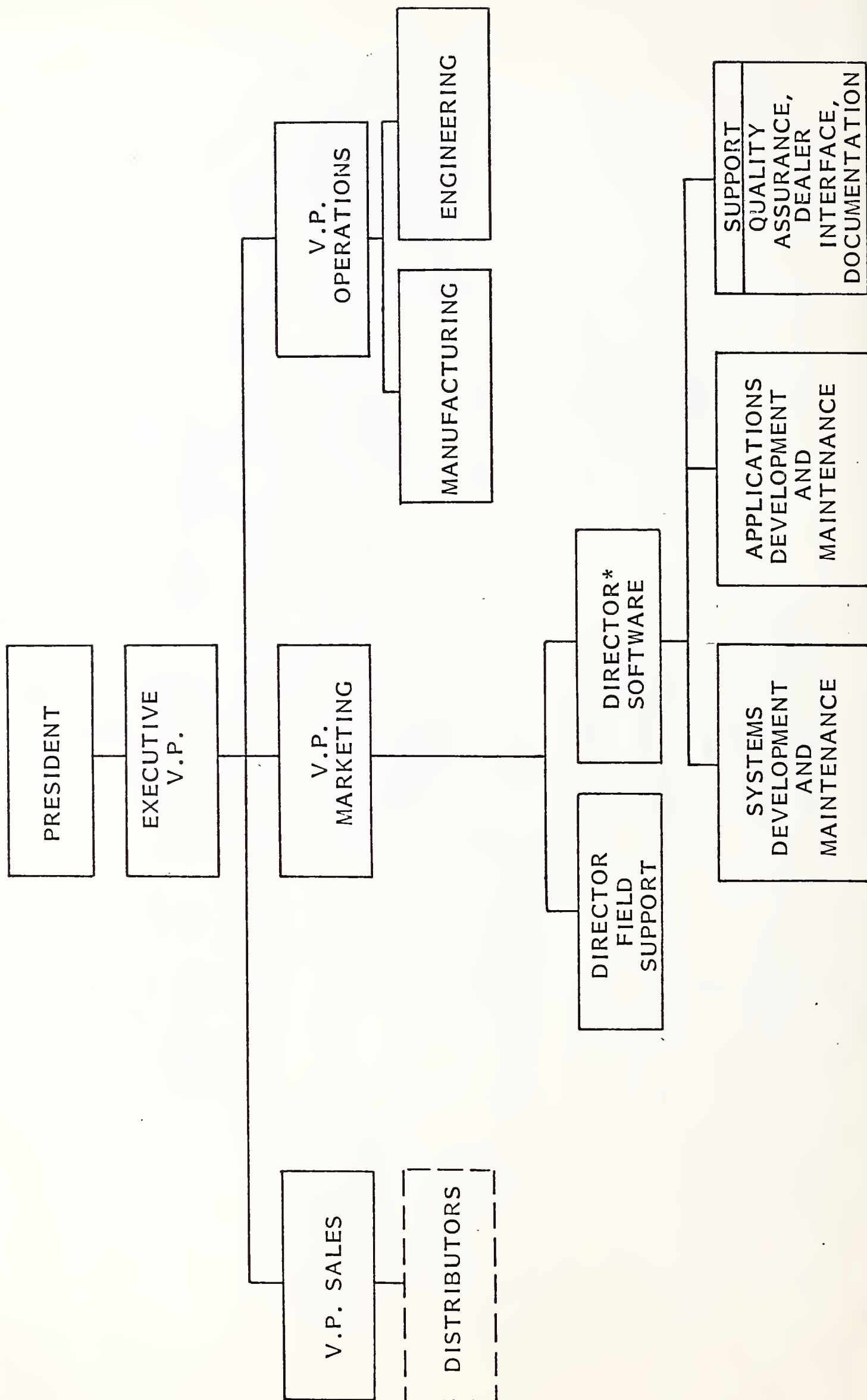
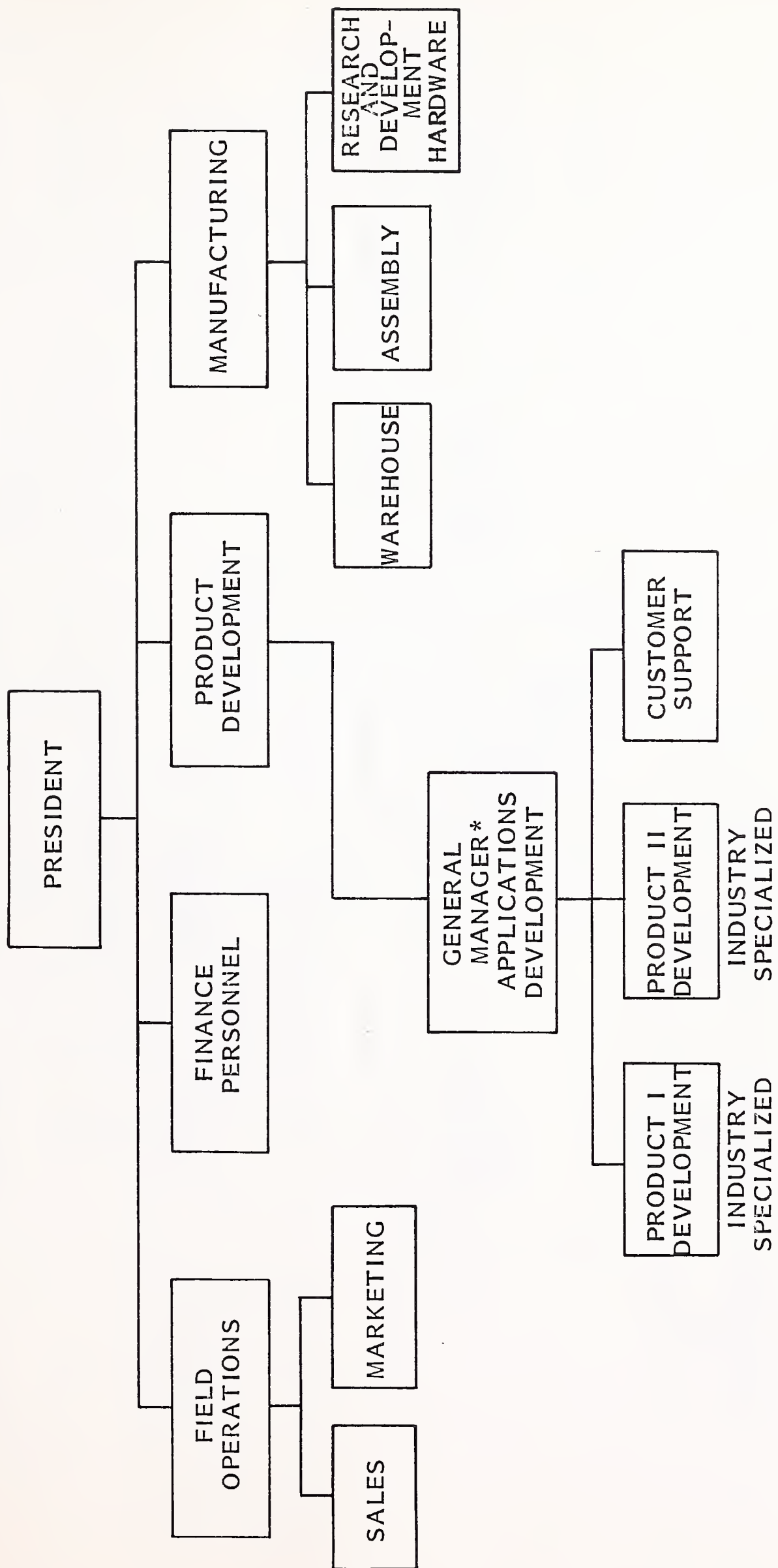


EXHIBIT C-16

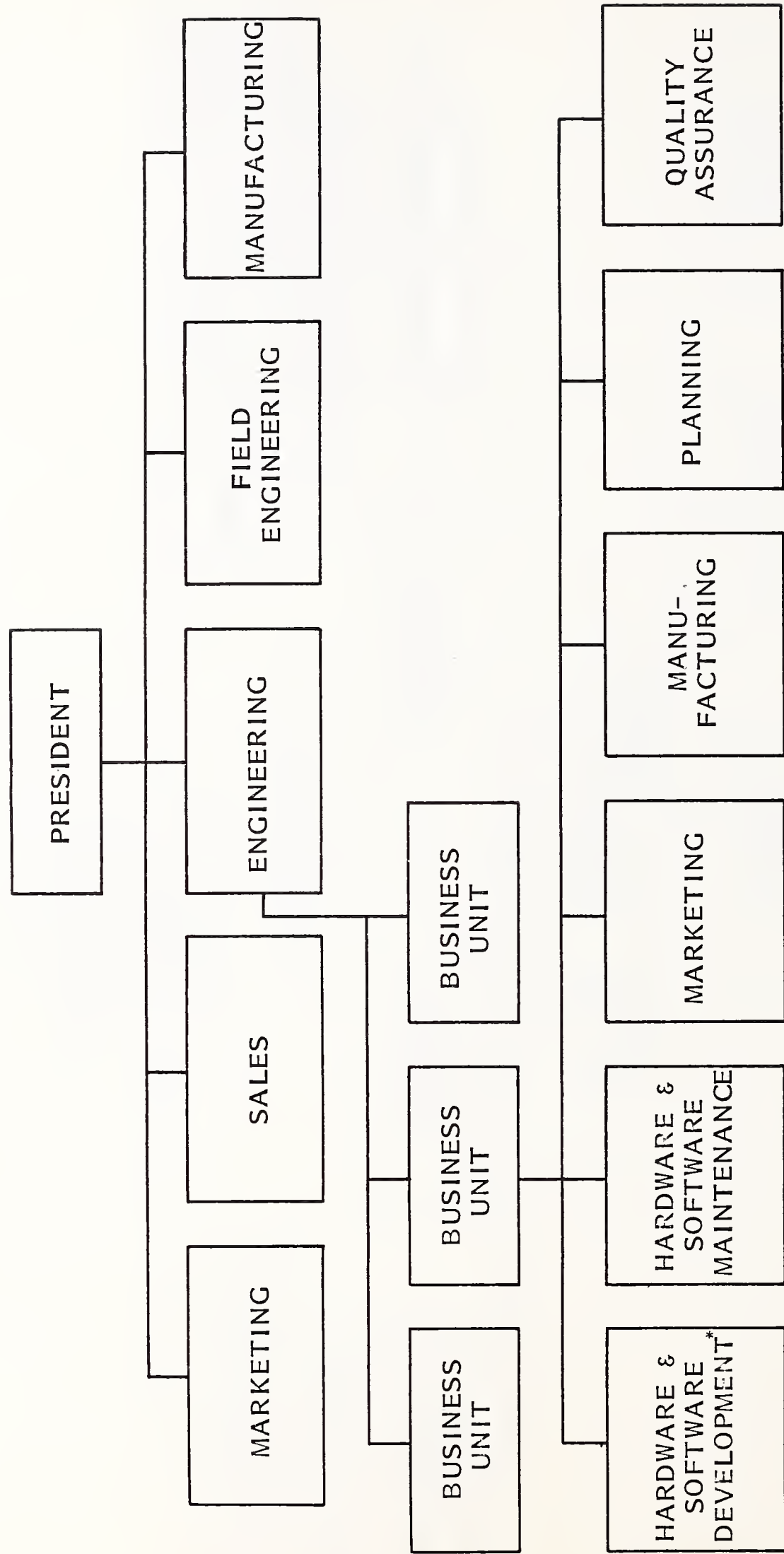
COMPANY P



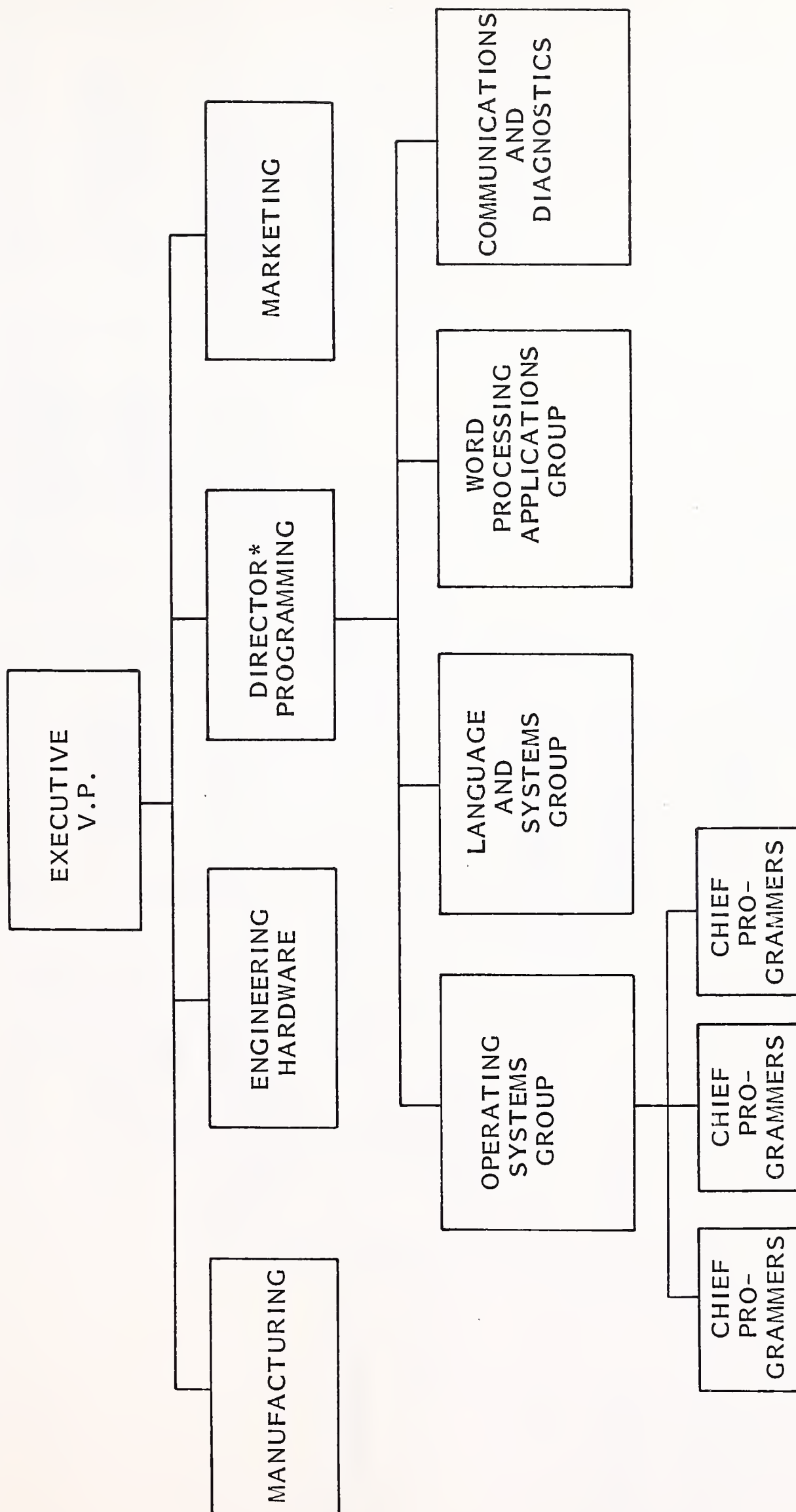
* INTERVIEWEE

EXHIBIT C-17

COMPANY Q



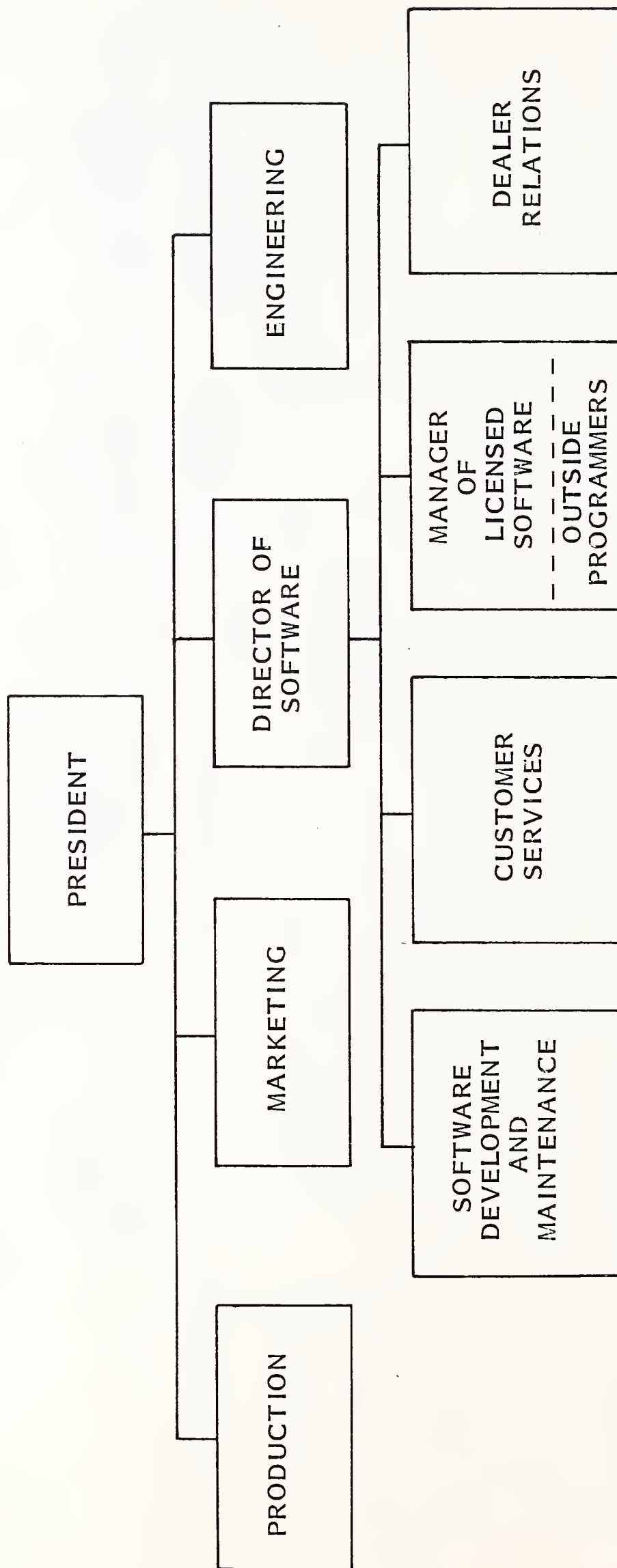
*INTERVIEWEE



* INTERVIEWEE

EXHIBIT C-19

COMPANY S



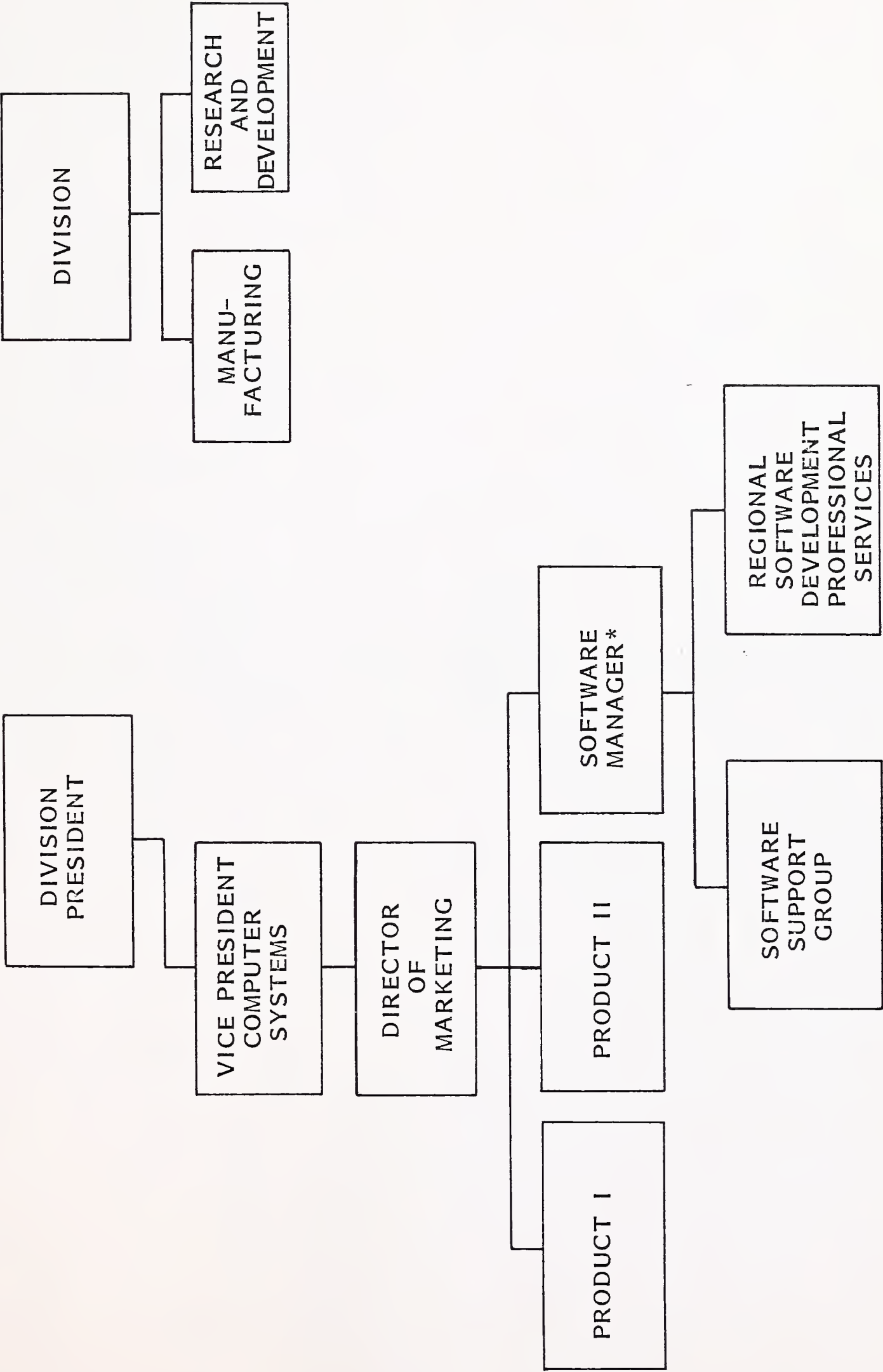
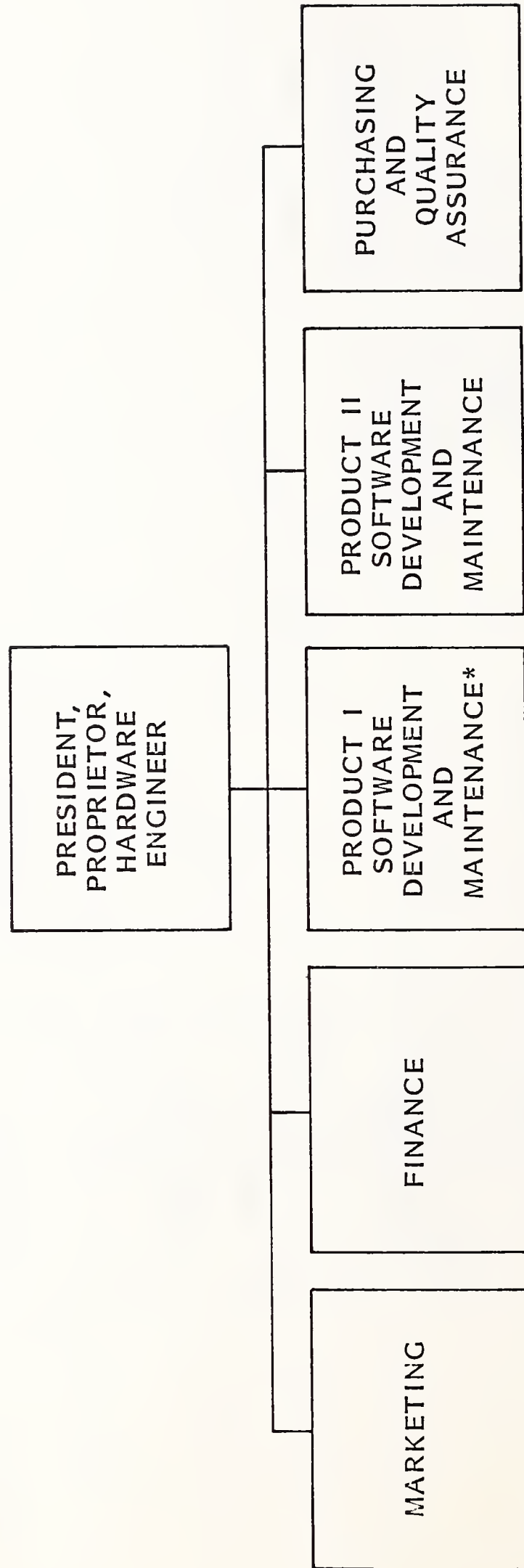


EXHIBIT C-21

COMPANY U



APPENDIX D: QUESTIONNAIRE

PRACTICES AND PROCEDURES
FOR THE DEVELOPMENT OF SOFTWARE

1. Company Categorization

a. Company type

- ☐ Hardware Computer Manufacturer
- ☐ OEM Integrator
- ☐ System Integrator
- ☐ Software Vendor

b. Software Category

- ☐ Systems
- ☐ Communications
- ☐ Applications
- ☐ Maintenance Only

c. Software Development Group Type

- ☐ New Start-up Company
- ☐ New Division or Group in Existing Company
(1 to 3 years)
- ☐ New Product in Existing Company
(3 to 5 years)

d. Company Computer Size Orientation

- ☐ Small (\$20,000 to \$100,000 mini-computer system)
- ☐ Medium (> \$100,000 but < IBM 370-158)
e.g., Tandem
- ☐ Large (IBM 370-158 or larger)
- ☐ Communications

2. Development Organization

- a. Where and to whom does the software development group report in the organization and what are the relationships to other functions? (An organization chart is required here.)

NOTE: Functions should include:

- a) Business Planning
- b) Market Planning
- c) Product Planning
- d) Marketing
- e) Sales
- f) Hardware Development
- g) Marketing Support
(of Field Sales)
- h) Software Maintenance
- i) Other (define)

2. b. Describe the organization structure for system and applications software development and maintenance. (Include organization chart for software development and maintenance group, if necessary.) Are they separated functionally?

☐

Yes

☐

No

2. c. Please separate your development/maintenance personnel into the following categories.

Category	Analysts	Programmers	Total
Development			
Maintenance (Sustaining)			
Total			

2. d. Describe your organizational strengths and weaknesses:

Strengths:

Weaknesses:

1) _____	1) _____
_____	_____
2) _____	2) _____
_____	_____
3) _____	3) _____
_____	_____

- e. Please rate the quantity and effectiveness of communications between software development and other departments. Use a scale of 1 to 5, where 1 is the smallest quantity and least effective, and where 5 is the highest quantity and most effective.

<u>Interface to Department</u>	<u>Quantity of Communications</u>	<u>Effectiveness of Communications</u>
Business Planning	_____	_____
Market Planning	_____	_____
Product Planning	_____	_____
Marketing	_____	_____
Sales	_____	_____
Hardware Development	_____	_____
Marketing Support (of Field Sales)	_____	_____
Software Maintenance	_____	_____
Other (define)	_____	_____
	_____	_____

- f. What suggestions do you have for interdepartmental communications improvement?

3. Software Costs:

* Answer a and b; or c (Procure the requested or most usable information.)

- *a. What were the software development expenditures as a percent of company revenues for the last fiscal year?

 %

- *b. What was the software maintenance expense as a percent of company revenue for the last fiscal year?

 %

- *c. What is the software development/maintenance expense as a percent of company revenue?
-
- %

** Answer d and e; or f.

- **d. What is the ratio of software development expense to hardware development expense?
-

- **e. What is the ratio of software maintenance expense to hardware development expense?
-

- f. What is the ratio of software development/maintenance expense to hardware development expense?
-

- g. How much is spent per employee on software development and maintenance?

1) Development \$ 2) Maintenance \$ 3) Total \$

- **h. For specific products, please answer the following questions.

	<u>Products</u>	<u>Development Expense</u>	<u>Year Started</u>	<u>Year Completed</u>	<u>Person-Years of Effort Expended</u>	<u>Lines of Code Produced</u>
1.	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>
2.	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>
3.	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>
4.	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>
5.	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>

4. Software Quality Control

- a. How do you measure the quality of your software development?

- b. What type of quality control mechanisms do you employ?

- c. Do you use programming standards?

☐ Yes ☐ No (If "No," go to question 4. g.)

- d. If you have standards, how do you rate their effectiveness? Use a scale of 1 to 5, where 1 is the least effective and 5 is the most effective.

_____ (Rating)

- e. If you have standards, how would you rate the level of adherence to those standards? Use a scale of 1 to 5, where 1 indicates the lowest level and 5 the highest.

_____ (Rating)

- f. If you have standards, how do you measure the level of adherence to those standards?

4. g. How do you measure software development success and failure?

1) Success: _____

2) Failure: _____

h. What software development successes and failures have you experienced within the last five years?

i. What procedural improvements would you recommend for increasing software product quality?

5. Software Development Future Issues

- a. What software development methodology do you envision being available by 1985?

- b. What software development methodology changes do you expect to implement by 1985?

6. Describe the product or type of product for which you develop software.

Y-54's

ORGANIZING FOR EFFECTIVE SOFTWARE DEVELOPMENT

ABSTRACT

INPUT conducted a survey of twenty-one computer manufacturers to describe and evaluate their software development procedures for small- and medium-size free-standing systems. In this report, the effectiveness of different organizational structures is evaluated. Interdepartmental interfaces and communications are also examined. Software development, maintenance costs and productivity are studied. The impact of programming standards and quality control mechanisms on productivity is analyzed. Other issues covered include the measurement of software development success and the future of software development.

